CRANFIELD UNIVERSITY

HYONDONG OH

TOWARDS AUTONOMOUS SURVEILLANCE AND TRACKING BY MULTIPLE UAVS

SCHOOL OF ENGINEERING
DIVISION OF ENGINEERING SCIENCES

PhD

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

Division of Engineering Sciences,
Centre for Cyber-Physical Systems

PhD Thesis
Academic Year 2010-2013

Hyondong Oh

Towards Autonomous Surveillance and Tracking by Multiple UAVs

Supervisor:
Professor Antonios Tsourdos

Nov 2013

Abstract

This research investigates the use of small and low-cost UAVs (Unmanned Aerial Vehicles) for autonomous aerial surveillance, which aims to identify and continuously track suspicious vehicles and disguised threats in the ground traffic. Since typical ground traffic in an urban environment is quite dense and involves numerous vehicles, achieving this surveillance capability by a single mobile platform is unlikely to be feasible in many aspects. In particular, due to physical constraints, it might be difficult for one UAV to cover large areas simultaneously, which is often critical to mission success in a rapidly changing environment. Besides, in order to obtain accurate information of ground traffic, a single UAV platform will need to rely on sensors which are expensive yet vulnerable to the failure of the platform or sensing block by obstacles. Using multiple UAVs with relatively cheap aboard sensors with information fusion techniques enhancing sensing accuracy could resolve above issues of a single platform without significantly increasing an operational cost.

Therefore, this thesis deals with the surveillance application of multiple airborne sensor platforms endowed with an appropriate level of autonomous decision making to support human operators. A group of UAVs become multiple mobile sensor platforms, and tasks/routes of each UAV need to be efficiently and optimally planned to cooperatively achieve mission objectives. Efficient and sophisticated algorithms for data acquisition/analysis, information fusion, path planning and formation reconfiguration ensuring feasible and safe cooperation, and decision making for cooperative missions are essentially to be developed, in order to take advantage of multiple aerial sensing sources for surveillance. Among various techniques for autonomous surveillance as listed above, this thesis seeks to develop and (partly) integrate some of important components: search route planning, behaviour identification/recognition, and moving target tracking, while examining benefits and drawbacks of using multiple UAVs. A particular focus is on multi-sensor management and information fusion in consideration of physical constraints of the platform and strict real-time requirements of the applications in uncertain and dynamic environments.

This thesis firstly proposes a road-network search planning algorithm by

which UAVs are able to efficiently patrol every road identified in the map. A mixed integer linear programming problem (MILP) is formulated to find an optimal solution minimising a total flight time, while accommodating physical constraints of the UAV with the Dubins path. To overcome the computational burden of the MILP, an approximation approach is also proposed. By running Monte Carlo simulation with the randomly generated maps, an efficient UAV team size and path planning method is examined. Secondly, this thesis proposes a behaviour recognition methodology for ground vehicles moving within road traffic to identify abnormal behaviour. Ground vehicle behaviour is first classified into representative driving modes, and string pattern matching theory is applied to detect suspicious behaviours in the driving mode history. Moreover, a fuzzy decision making process is developed to systematically exploit all available information obtained from a complex environment considering spatiotemporal environment factors as well as several aspects of behaviours. Lastly, to achieve continuous tracking of detected suspicious vehicles for closer and higher-resolution surveillance data, this thesis proposes several coordinated standoff tracking guidance algorithms using multiple UAVs. The effect of the improved target estimation accuracy on the tracking guidance performance is also examined using roadmap information and sensor fusion techniques.

From this thesis, it can be identified that following aspects need to be carefully considered to realise autonomous surveillance using multiple UAVs: i) how many UAVs/sensors would be enough to perform a mission in terms of efficiency, estimation accuracy and guidance performance, ii) information gathered by UAVs only is enough, or domain knowledge (local context and past experience) might be additionally required, iii) communication structure between UAVs, and iv) computation time. The proposed autonomous surveillance system utilising multiple UAVs is expected to greatly increase the amount of area that can be continuously monitored, while reducing the number of human operators and their workload required to analyse surveillance data and respond to identified targets.

To
the Lord of all creation and my beloved family

Acknowledgements

I would like to thank my supervisor Professor Antonios Tsourdos for giving me a great opportunity to undertake my PhD research. His considerate support and concern helped me to settle down in a new research environment comfortably right from the start of my study at Cranfield. In addition, throughout my PhD, his constant inspiring and practical guidance for the research has contributed significantly to a successful completion of my research work. I am deeply indebted to him for a positive influence on my academic life.

I am thankful to Dr. Hyo-Sang Shin and Dr. Seungkeun Kim for not only their insightful and critical guidance regarding my research but also being a good model of researcher and mentor to me in many ways. I would like to also thank internal/external examiners Professor Rafal Zbikowski and Dr. Antony Waldock, BAE systems, for taking time out of their busy schedule to review my thesis and providing valuable suggestions to improve the quality and presentation of my work. In addition, I would like to thank other academic members in Autonomous and Intelligent Systems (AIS) group Professor Brian White, Dr. Al Savvaris, and Dr. Peter Silson for their guidance and encouragement in completing several research projects during my PhD. I would like to thank my former advisor Professor Min-Jea Tahk at KAIST, South Korea for his practical advices on both my academic career and life.

I am grateful to my colleagues and friends Dr. Weizhong Fei, Dr. Jiyoung Choi, Mr Rober Ioiart, Mr Chang How Lo, Mr Cedric Leboucher and Mr David Wall with whom I enjoyed the time at Shrivenham (Defence Adademy of the UK) and Cranfield campus. I would like to also thank Pastor Gyesang Jeon and Oxford Mission Church (OMC) members. I have gone through all of my ups and downs in my physical and spiritual life with them in the UK; they have become more than a family to me. I would like to express my deep sense of gratitude with all my heart to my parents and sister. Their endless support and encouragement have enormously helped me to complete my PhD study and to keep the momentum moving forward as ever. Last, but not the least, I would like to give thanks to my wife Jiyeon Yang who came to the UK leaving many things behind to support me. Without her love, patience and smile, I could not make it this far.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The large number of UAV (Unmanned Aerial Vehicle) applications has proliferated within the last two decades with the fielding of Global Hawk, Pioneer, Pathfinder Raven, and Dragon Eye UAVs [4]. The operational experience of UAVs has proven that their technology can bring a dramatic impact to both military and civilian applications [5, 6]. Groups of small and low-cost UAVs are of special interest due to their ability to coordinate simultaneous coverage of large areas, and co-operate to achieve common goals [7, 8]. The intelligent and autonomous cooperation of multiple small UAVs operating in a team offers revolutionary capabilities: improved situation awareness with global and/or local information within a given time and resource [4]; higher survivability and mission success rate; significant reductions in manpower and risk to humans [9]; the ability to perform missions in hostile, hazardous and geometrically complex environments [10]; and cost efficiency. Some of above capabilities are unique to multiple UAVs or could be enhanced by the intelligent use of UAVs with planning and information fusion. Specific applications under consideration for groups of cooperating UAVs include, but are not limited to, border patrol and surveillance [11, 9], search and rescue [12], airborne police law enforcement [13], ground vehicle convoy [14], mapping and environmental monitoring [15, 10].

In particular, surveillance, and subsequent tracking, of moving ground targets of interest has become one of the primary UAV capabilities; yet has not been completed, as mentioned in [5] for a military perspective, among others:

> "For the next 25 years, the Department of Defense will focus the labs and industry on the following mission areas: intelligence, surveillance and reconnaissance (ISR), ..."
>
> "Unmanned aircraft (UA) have an established and growing track record supporting the ISR mission area. The concept of using miniature UA

> to conduct collection against weak signals or obtain high resolution results is an emerging capability that deserves increased emphasis."

In general, surveillance represents the monitoring of the behaviour, activities, or other changing information, usually of people and vehicles for the purpose of influencing, managing, directing, or protecting [16]. Aerial surveillance is the gathering of information, usually visual imagery or other sensor data, from an airborne vehicle. Programs such as the Heterogeneous Aerial Reconnaissance Team (HART) developed by Defence Advanced Research Projects Agency (DARPA) have dealt with automating the aerial surveillance process [1]. In this program, multiple automated, self-directing drones automatically patrol a city and track suspicious individuals, reporting their activities back to a monitoring station, as illustrated in Fig. 1.1. In the civilian domain, various organisations such as the U.S. Department of Homeland Security and Transportation are in the process of testing UAVs to patrol the skies for the purposes of critical infrastructure protection, border patrol, transit monitoring, traffic control, and general surveillance of the population, to be used by police forces [17, 18]. This aerial surveillance system is required to increase an overall knowledge about surrounding environments and to take proactive measures for preventing safe-critical events from happening.

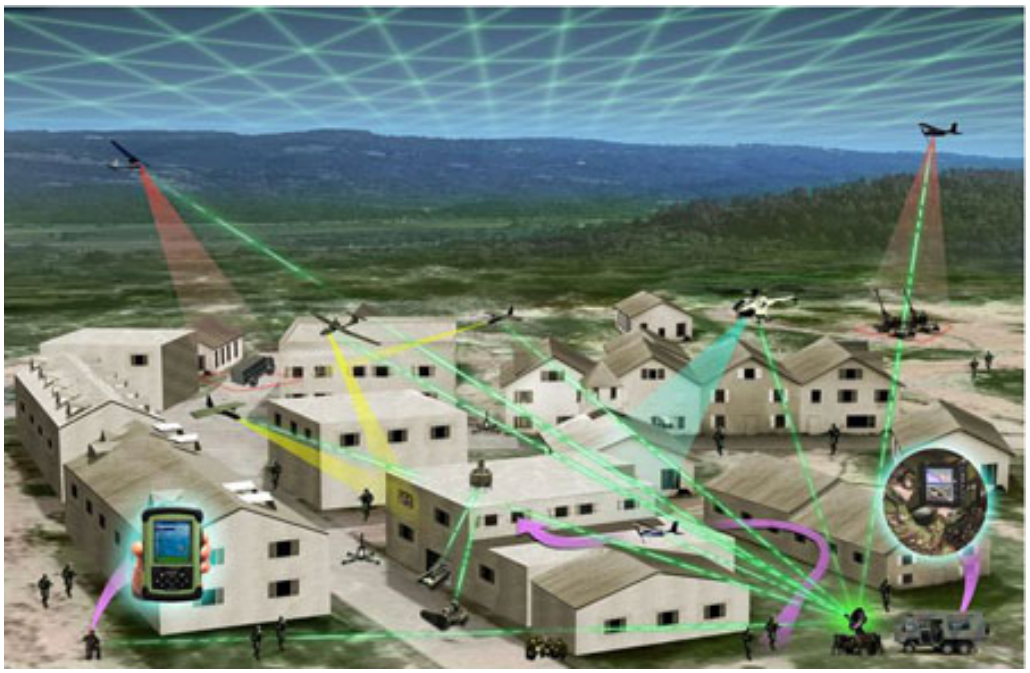


Figure 1.1: Concept of DARPA's Heterogeneous Aerial Reconnaissance Team [1]

1.2 Background

UAVs for Surveillance

The traditional approach for surveillance before the advent of UAV technology is to use a single manned airplane, space assets, or ground based systems. Manned airplanes are typically large and expensive; dangerous or hostile environments and operator fatigue from repetitive tasks can potentially threaten the life of a pilot. Space assets such as satellites are also very expensive and restricted to use. Even though space assets are usually employed globally, some satellites cannot conduct *unwarned* information collection since adversaries might know when they will come above the horizon to deny collection opportunities [5]. Meanwhile, UAV advantages of being able to operate in both flexible time and space and no human on-board provide significant opportunities to achieve an *unwarned* collection capability, focusing resources on the current problem. Compared to ground surveillance sensors such as loop detectors and video cameras positioned at fixed locations [19, 20], UAV aerial sensing can provide better coverage to survey large areas at a high speed without being confined to prescribed ground navigation routes. In addition to these, UAVs can obtain timely information as well as respond to the identified situation and target quickly from the air, which is the most critical requirement in case of emergencies such as accidents, natural disaster, oil leaks and etc. Therefore, there is a strong need to use UAVs for surveillance missions.

High-Level Behaviour Analysis for Surveillance

The UAV surveillance system inherently requires target acquisition and detection techniques through area patrol and behaviour analysis of gathered live or recorded surveillance data. For instance, UAVs with an onboard visual/ground MTI (moving target indicator) radar provide accurate estimation of a large number of moving targets [2, 21, 22] and consequently allow rapid and consistent detection of possible threats. When the surveillance data is combined with visual mosaic techniques, target trajectories and motions can be represented on the map, and hence can increase the operator's situational awareness significantly. However, the operator still needs to analyse and interpret the resulting surveillance data, and constructs a picture of events in order to detect suspicious behaviours.

Within the ground traffic the most difficult challenge is to recognise potentially dangerous vehicles disguised as legitimate traffic, which can entail weapons and personnel smuggling, target reconnaissance by terrorists, roadside bomb planting, suicide bombers, etc. Most of these activities of the dangerous vehicles are characterised by occasional deviations from motion characteristics of legitimate traffic. This behaviour analysis requires several highly-skilled human operators, which is expensive and unsustainable under deluge of data and information. Besides, current research using UAVs has focused on the development of the separate algorithms, e.g. sensing and modelling [23, 24], multi-sensor management and information fusion [25, 26], or guidance and control [27, 28] required for the ground traffic surveillance. However, these research works have rarely been integrated into a system to aid situation assessment. Hence, there is another strong need to develop an integrated high-level analysis algorithm to process target information and detect anomalous behaviours to reduce human operator's workload.

Ground Target Tracking with Information Fusion

Once a target is identified through airborne surveillance, further monitoring might be required to obtain closer and higher-resolution surveillance data for positive identification, by approaching more closely and continuously tracking it. This tracking task can be regarded as part of surveillance in a wider sense despite it can be separately used for other missions. This is not an easy task due to unknown target manoeuvres, as well as kinematic constraints of UAVs. For this, a standoff tracking concept is introduced, which is to keep a certain distance (termed standoff distance) from the moving target resulting in orbiting around the target, in order to track without being noticed and keep it within the field-of-view (FOV) of the sensor [14, 29]. Coordinated standoff tracking is also proposed by distributing a team of UAVs on the same standoff orbit, in order to obtain better estimation accuracy [30], as well as more robust performance in case of sensing failure of individual sensors or obscuration of the target [28]. Although considerable approaches on this problem have been developed, there are still many aspects yet to be improved, such as robustness and performance of the guidance algorithm.

In this regard, note that the performance of this tracking guidance is strongly

coupled with sensing and estimation capability. Since ground target tracking is a challenging problem, all available information along with individual sensor data themselves need to be exploited; information fusion could be used to enhance the performance of the guidance as well as the estimation significantly. Multi-sensor fusion using other UAV's measurements and estimates for the same target with contextual information could be one of possible approaches to implement information fusion. In this thesis, information fusion is used as broad terminology for acquiring, processing, and intelligently combining information where the tracked entities are associated with environmental, doctrinal, and performance constraints and then assessed in view of multi-perspectives [31]. In particular, in many applications for ground target tracking, since the majority of ground vehicles are moving on road networks, such road-map information can be used for improving the quality of estimation [32, 33, 34]. However, these aspects have rarely been considered in the standoff tracking problem.

Multi-Target Tracking Using Multiple UAVs

Furthermore, the previous research on standoff tracking has focused on the single target tracking problem. In case that multiple moving ground vehicles are identified as targets of interest from surveillance systems, it is required to develop strategies on how to deploy multiple UAVs to persistently follow them. Keeping all targets in the FOV of the sensor gets more difficult due to acceleration, decelerations, and possibly deliberated evasive manoeuvres. Multiple platforms could be used, anticipating different turns at intersections and effecting hand-off of a target from one UAV to another as needed to ensure reliable tracking [4]. Various different methodologies have been developed for the multi-target tracking using multiple ground [35, 36] or aerial vehicles [37, 38]. However there is little research available regarding on multiple or group target tracking in the context of standoff tracking guidance, considering uncertain and dynamic environments and UAV sensing capability.

Integrated Cooperative Surveillance System

For the aforementioned autonomous surveillance problem, a group of UAVs become multiple airborne sensor platforms, and consequently tasks and routes of each UAV need to be efficiently and optimally planned in order to coopera-

tively achieve their mission. Directions i) to search or monitor particular areas, ii) to patrol along certain strategic roads and infrastructures, and iii) to track potential targets obtained by the behaviour analysis will be typical instructions, with a team of UAVs reacting to produce cooperative behaviour autonomously. The challenges in practical cooperative decision and control problems are driven depending on the specific requirements of the proposed missions [39, 40]. In particular, the main challenges of migration from current manned or ground based systems to UAVs are due to the physical (dynamic, kinematic, and communication) constraints of the platform and the strict real-time requirements of the applications under uncertain environments [7]. Thus, efficient algorithms for data acquisition/analysis, information fusion, path planning and formation reconfiguration which ensure feasible and safe cooperation between UAVs, and decision making for cooperative missions are essentially to be developed and integrated, in order to take advantage of multiple aerial sensing sources.

1.3 Aim and Overview of the Thesis

The goal of this thesis is to examine the use of a team of small and low-cost UAVs for autonomous aerial surveillance, which aims to identify and continuously track suspicious vehicles and disguised threats in the ground traffic, and then notify human operators about potentially dangerous vehicles in the area of interest. Since typical ground traffic in an urban environment is quite dense and involves numerous vehicles, achieving these surveillance capabilities by a single sensor platform is unlikely to be feasible. In particular, due to physical constraints (minimum turning radius, speed, and sensing field-of-view/range), it might be difficult for a single UAV to cover large areas simultaneously, which is often critical to mission success in a rapidly changing environment. It would be also hard to obtain sensor measurements of targets from different angles to have better understanding of them, particularly against fast moving ones. Besides, in order to obtain accurate information of ground traffic, a single UAV platform will need to rely on sensors which are expensive yet vulnerable to the failure of the platform or sensing block by obstacles. Using multiple small UAVs with relatively cheap aboard sensors with information fusion techniques enhancing sensing accuracy could resolve above issues of a single platform without significantly increasing

an operational cost.

Therefore, this thesis presents the surveillance application of multiple airborne sensor platforms (i.e. UAVs equipped with onboard sensors) endowed with an appropriate level of autonomous decision making, while examining benefits and drawbacks of using low-cost multiple UAVs. The UAV team should be able to decide its collective and individual actions using different pieces of information, classifying what information is necessary, recommended or just desirable. To achieve this under a deluge of data and information from complex scenes, multi-sensor management and information fusion are extensively addressed in consideration of physical constraints of the platform and strict real-time requirements of the applications. This allows mobile sensor platforms to intelligently obtain and process target information for anomalous behaviour monitoring.

Among various techniques for autonomous surveillance, this thesis seeks to develop and (partly) integrate following necessary components, exploiting geographical, spatial and temporal information associated with operational environments, as listed:

- **Road-network search route planning** algorithms for multiple UAVs to efficiently patrol along certain strategic roads of interest (Chapter 3);
- **Ground moving target estimation** algorithms with information fusion: road-map assisted filtering, sensor fusion and optimal smoothing techniques (Chapter 4);
- **Behaviour recognition** algorithms to identify suspicious beahviour or potential threats within ground traffic (Chapter 5); and
- **Ground moving target tracking guidance** algorithms to follow moving ground targets closely as well as obtain accurate information of target motion and intent (Chapter 6 and 7).

Each part is investigated in corresponding chapters as indicated, and the performance of partially integrated algorithms is examined by numerical simulations with realistic scenarios using real VMTI sensor measurements and an off-the-shelf traffic simulation program.

Figure 1.2 shows the relations between the proposed algorithms for cooperative aerial surveillance and target tracking. Ground vehicle motions are first observed and relevant state estimates (such as position, velocity, and acceleration) are computed using a filtering technique by individual UAVs. State estimates obtained from a team of UAVs are transmitted to a ground mission control station, and followed by data/information fusion to improve the estimation accuracy. Processed accurate target information and behaviour recognition results for suspicious vehicles are then used for decision making.

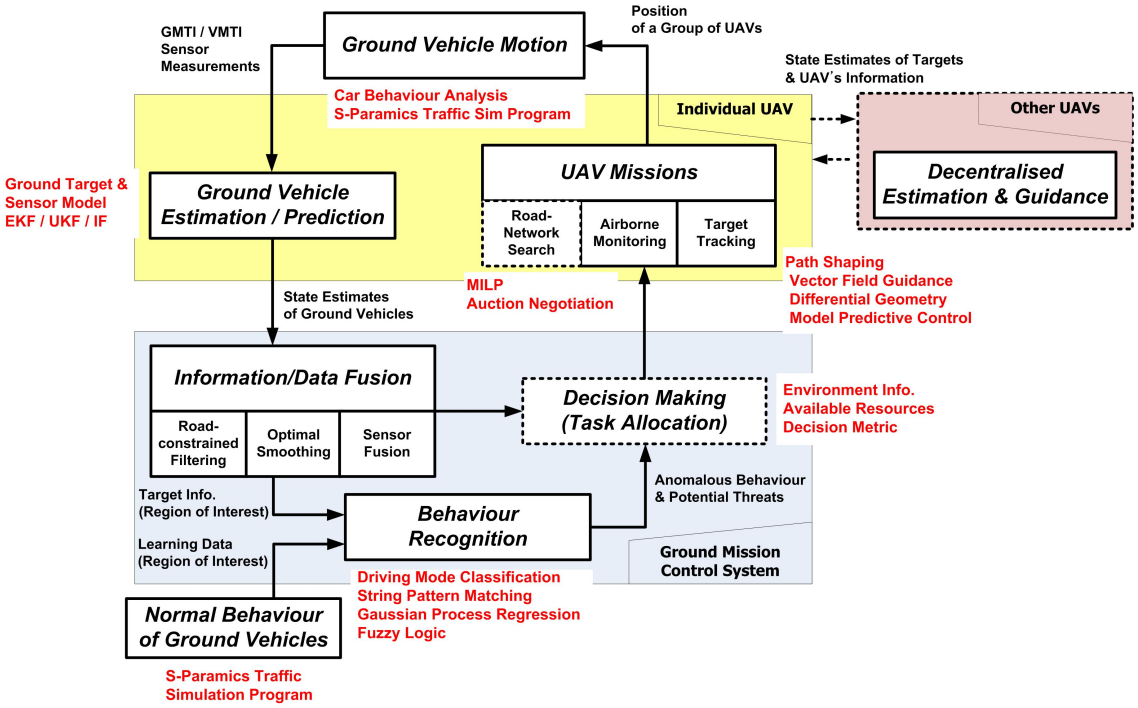


Figure 1.2: Block diagram of the proposed algorithms

Airborne monitoring and behaviour recognition of the ground traffic involves a variety of processing modalities defined by different level of abstraction of ground vehicle motion characteristics. The first type of modality is a range of vehicle trajectories corresponding to legitimate behaviour, e.g. moving in a straight line within a range of speeds, turning at a range of rates and obeying traffic rules. The other type of modality is clear patterns of alarming motion, e.g. repeated cruising in a sensitive area, unusual stopping and unexpected disembarking and unloading in non-designated areas. These modalities would have to rely on a variety of sensors, but this research focuses solely on the sensors for kinematic

behaviour of ground vehicles, e.g. airborne visual/ground moving target indicator (MTI) radar. On a low level of abstraction, recognition of kinematic behaviour is needed, whereas recognising the modality for a given vehicle and deciding whether to follow the vehicle is a higher level decision making. All these problems require techniques to plan and manage just-in-time decision making. Also, the definition of normal behaviour must be based on past experience and local context, as legitimate traffic and driver's behaviour differ depending on the region. These aspects are carefully dealt with in this thesis.

The decision making, represented as dashed line in Fig. 1.2, on how to optimally deploy multiple UAVs is tackled within individual tasks (road-network search, airborne monitoring, and target tracking) in this thesis. Assigning UAVs to a set of heterogeneous tasks/missions (i.e task allocation) is assumed to be done by human operators in the mission control system. The mission control station will have dedicated operators reviewing and approving the automated decision and ensuring the right information/command get delivered to the right UAV, using extensive automation tools to facilitate the task of managing the resources. For a fully autonomous surveillance system, autonomous decision making algorithm needs to take the role of human operators; this aspect remains as future work of this thesis. Some of the relevant research can be found in [39, 41, 42, 43].

Note that some of algorithms are performed in a decentralised manner [20, 40](e.g. Section 4.6.4 for sensor fusion, Section 6.2 and 6.4 for tracking) considering real-world communication and computation constraints, while exchanging information and data between UAV via communication links. Besides, road-network search mission is tackled in the context of multi-UAV route optimisation without considering a target model and behaviour for patrolling purpose only.

1.4 Summary of Contributions

The contributions of this thesis are aimed at largely three classes of a surveillance problem: (i) coordinated road-network search strategies using the Dubins paths, (ii) efficient behaviour classification and recognition of ground vehicles, and (iii) standoff tracking guidance algorithms with the improved estimation accuracy using multiple UAVs and information fusion, and their performance comparison

and analysis.

Firstly, this thesis proposes a road-network search planning algorithm by which multiple UAVs are able to efficiently patrol every road identified in the map, as presented in Chapter 2 and 3. A mixed integer linear programming (MILP) problem is formulated to find an optimal solution minimising a total flight time. In order to accommodate the physical constraints of the UAV in the search route design, this study uses the Dubins path [44], which is the shortest path connecting two configurations under a bound on curvature. Both CCC (circular-circular-circular) and CSC (circular-straight-circular) type of the Dubins paths are exploited to precisely cover a densely distributed road environment; in particular, CCC path is derived by using differential geometry, which has not been done in the literature. Besides, to overcome the computational burden of the MILP algorithm, an approximation approach, based on nearest insertion and auction negotiation, is newly proposed. By running Monte Carlo simulation, an efficient UAV team size and path planning method is advised for the specific road-network search mission.

Secondly, the contributions to the behaviour recognition problem are as follows, presented in Chapter 4 and 5. This thesis proposes a systematic and integrated approach for the autonomous recognition of ground traffic behaviour. It includes following techniques: ground target tracking, sensor fusion and trajectory refinement, trajectory classification, behaviour recognition. One of main contributions is to propose a classification methodology which simplifies a complicated driving behaviour as a sequence of integers. A simple but effective detection scheme for irregular driving behaviours is proposed, based on string matching theory. An anomaly detection algorithm is also proposed using a statistical learning approach with domain knowledge given by road-map information. Although the proposed string matching and anomaly detection algorithms can provide a measure of suspicious behaviour, additional information needs to be considered to finally confirm the characteristic of behaviour, in order to avoid frequent false alarms. Therefore, a fuzzy rule-based decision making is developed to accommodate several aspects of behaviour as well as taking into account the spatiotemporal environment factors, thus providing a more effective level of alert to the operator monitoring complex scenes.

Lastly, this thesis proposes several coordinated standoff tracking guidance

algorithms, some of which is in conjunction with a road-map assisted filtering and sensor fusion technique, as presented in Chapter 6. These are: (i) two-phase orbit approach using path shaping, (ii) decentralised vector field guidance using adaptive sliding mode control, (iii) differential geometric guidance, and (iv) nonlinear model predictive coordinated standoff tracking (NMPCST). Compared to existing methods, the proposed methods have various innovative aspects such as utilising path shaping to adjust the angular separation between UAVs operating at a constant speed, combining decentralised estimation and guidance considering different information/communication networks, and rigorous stability analysis with differential geometry framework. The NMPCST is of special interest. Unlike the decoupled control structure of the existing works, the NMPCST seeks to find the heading and velocity control inputs simultaneously using a coupled structure, providing an optimal performance in terms of tracking and control efforts. In addition, in this model predictive approach that exploits future target trajectories, the prediction of the target movement plays an important role on tracking performance. Thus, this thesis investigates the effect of the improved estimation accuracy on the guidance performance. Moreover, for the case that multiple ground vehicles are identified as targets of interest, this study proposes a coordinated standoff tracking of groups of moving targets using multiple UAVs considering UAV and sensing constraints.

1.5 Organisation of the Thesis

This thesis is organised as follows. Chapter 2 presents the path planning algorithms for the UAV surveillance problem as a background of this thesis. Chapter 3 presents the road-network search strategies by which multiple UAVs are able to efficiently patrol the identified road-network. In Chapter 4, ground moving target tracking is described using target dynamics, a visual/ground MTI sensor model, and estimation algorithms. Chapter 5 proposes a systematic and integrated approach for the autonomous recognition of ground traffic behaviour. In Chapter 6 and 7, several innovative coordinated standoff tracking guidance algorithms are proposed to persistently follow a moving target or groups of targets, and the effect of improved estimation accuracy on the guidance performance is investigated. In Chapter 8, the thesis ends with conclusions and discussion of related future work.

1.6 Publications

The following papers are published in relation to this PhD research. Note that journa paper [J6] is first-authored by one of my research group members, but I contributed to it significantly in both algorithm development and writing.

1.6.1 Book Chapters

[B1] **H. Oh**, H.S. Shin, S. Kim, A. Tsourdos, and B.A. White. Cooperative mission and path planning for a team of UAVs. *Handbook of Unmanned Aerial Vehicles, Kimon P. Valavanis and George J. Vachtsevanos (eds)*, Springer, 2012. in press.

1.6.2 Journals

[J1] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Decentralised standoff tracking of moving targets using adaptive sliding mode control for UAVs. *Journal of Intelligent and Robotic Systems*, DOI:10.1007/s10846-013-9864-0, 2013. in press.

[J2] **H. Oh**, H.S. Shin, S. Kim, A. Tsourdos, and B.A. White. Airborne behaviour monitoring using Gaussian Processes with map information. *IET Radar, Sonar and Navigation*, 7(4):393-400, 2013.

[3] **H. Oh**, S. Kim, H.S. Shin, A. Tsourdos, and B.A. White. Behaviour recognition of ground vehicle using airborne monitoring of UAVs. *International Journal of Systems Science*, DOI:10.1080/00207721.2013.772677, 2013. in press.

[J4] **H. Oh**, D. Turchi, S. Kim, A. Tsourdos, L. Pollini, and B.A. White. Coordinated standoff tracking using path shaping for multiple UAVs. *IEEE Transactions on Aerospace and Electronic Systems*, 2013. in press.

[J5] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Coordinated road-network search route planning by a team of UAVs. *International Journal of Systems Science*, DOI:10.1080/00207721.2012.737116, 2013. in press.

[J6] S. Kim, **H. Oh**, and A. Tsourdos. Nonlinear model predictive coordinated standoff tracking of moving ground vehicle. *Journal of Guidance, Control, and Dynamics*, 36(2):557-566, 2013.

[J7] **H. Oh**, S. Kim, H.S. Shin, B.A. White, A. Tsourdos, and C.A. Rabbath. Rendezvous and standoff target tracking guidance using differential geometry. *Journal of Intelligent and Robotic Systems*, 69:389-405, 2013.

1.6.3 Conferences

[C1] **H. Oh**, S. Kim, J. Suk, and A. Tsourdos. Coordinated trajectory planning for efficient communication relay using multiple UAVs. *IFAC Symposium on Automatic Control in Aerospace (ACA)*, Wurzburg, Germany, 2013.

[C2] **H. Oh**, S. Kim, H.S. Shin, A. Tsourdos, and B.A. White. Coordinated standoff tracking of groups of moving targets using multiple UAVs. *Mediterranean Conference on Control and Automation*, Crete, Greece, 2013.

[C3] **H. Oh**, S. Kim, H.S. Shin, A. Tsourdos, and B.A. White. Fuzzy expert rule-based airborne monitoring of ground vehicle behaviour. *The UKACC International Conference on Control*, Cardiff, UK, 2012.

[C4] **H. Oh**, S. Kim, H.S. Shin, B.A. White, A. Tsourdos, and C.A. Rabbath. Rendezvous and standoff target tracking guidance using differential geometry. *International Conference on UAS (ICUAS'12)*, Philadelphia, USA, 2012.

[C5] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Road-map assisted standoff tracking of moving ground vehicle using nonlinear model predictive control. *American Control Conference*, Montreal, Canada, 2012.

[C6] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Decentralised road-map assisted ground target tracking using a team of UAVs. *9th IET Data Fusion & Target Tracking*, London, UK, 2012.

[C7] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Coordinated standoff target tracking of unmanned aerial vehicles using adaptive sliding mode control. *Research, Development and Education on UAS (RED-UAS 2011), Seville, Spain, 2011.*

[C8] **H. Oh**, S. Kim, A. Tsourdos, and B.A. White. Cooperative road-network search planning of multiple UAVs using Dubins paths. *AIAA Guidance, Navigation and Control Conference*, Portland, USA, 2011.

[C9] **H. Oh**, H.S. Shin, A. Tsourdos, B.A. White, and P. Silson. Coordinated road network search for multiple UAVs using Dubins path. *1st CEAS Specialist Conference on Guidance, Navigation and Control*, Munich, Germany, 2011.

Chapter 2

Path Planning for the UAV Surveillance and Tracking

This chapter presents path planning algorithms for the UAV surveillance problem as a background of this thesis, which will be used in the following chapters. The path planning for the UAV should be able to produce a feasible path whose end points are characterised by position (x, y) and orientation σ, together called a configuration $P(x, y, \sigma)$. The path planning is used differently from the trajectory planning which considers time as well. A feasible path is both flyable, meeting the kinematic and dynamic constraints and safe to fly without collision into each other or any other obstacles. Given initial and final configurations $P_s(x_s, y_s, \sigma_s)$ and $P_f(x_f, y_f, \sigma_f)$, respectively, maximum curvature κ_{max} (or minimum turning radius r_m) and the other constraints $\coprod$ such as obstacles, the path planner generates a path $r(t)$ with parameter t as:

$$P_s(x_s, y_s, \sigma_s) \xrightarrow{r(t)} P_f(x_f, y_f, \sigma_f), \quad \kappa(t) \leq \kappa_{max}, \text{ and } \coprod. \tag{2.1}$$

The main constraint for the UAV is its limited curvature or turn radius. This, together with producing paths that are as short as possible, requires the development of approaches that take these constraints into account. It is advantageous for the paths to have a shorter length as well as lower curvature differences between discontinuous points as possible in order to minimise flight time or energy consumption [45]. The techniques can then be extended to produce paths with longer lengths for the purpose of obstacle avoidance or to synchronise the arrival time without controlling the speed for autonomous vehicles.

In the literature, different approaches have been used to generate the feasible path satisfying constraints of the UAV. Chandler *et al.* [46] used Voronoi diagram to produce routes for the UAVs by minimising radar detection, and the routes were then refined to make them flyable. Rapidly-Exploring Trees (RRTs) is also used for collision free paths, with a path smoothing algorithm based on cubic spiral curves to satisfy the minimum turning radius constraints in [47]. Optimisation techniques are also used such as probabilistic methods [48], MILP [49], and genetic

algorithm [50, 51] to produce paths by optimising certain cost function. In these approaches, the final outcome is typically a route plan which satisfies certain constraints; if the route is refined by adding constraints to consider kinematics of the UAV, the resulted path is a series of lines and arcs, which is a subset of Dubins paths [44]. Dubins showed that the shortest path between two vectors in a plane and meets minimum bounds on turning radius is a composite path formed by the segments of line and circular arcs, term as the Dubins paths. A mathematical proof is provided by Dubins and further dealt with by others ([52, 53, 54, 55]). It is commonly accepted as a reasonably accurate kinematic model for aircraft motion planning problem ([56, 57]). Shanmugavel *et al.* [58] used the Dubins path set of curves as well as Pythagorean curves connecting the start and goal points for UAVs, based on differential geometry. Even though the path generation is straightforward, this approach uses iteration to satisfy the constraints. They also proposed the use of clothoid arcs which have ramp curvature profile providing a smooth curvature transition between the arcs and the line segment, in order to address the path discontinuities [59].

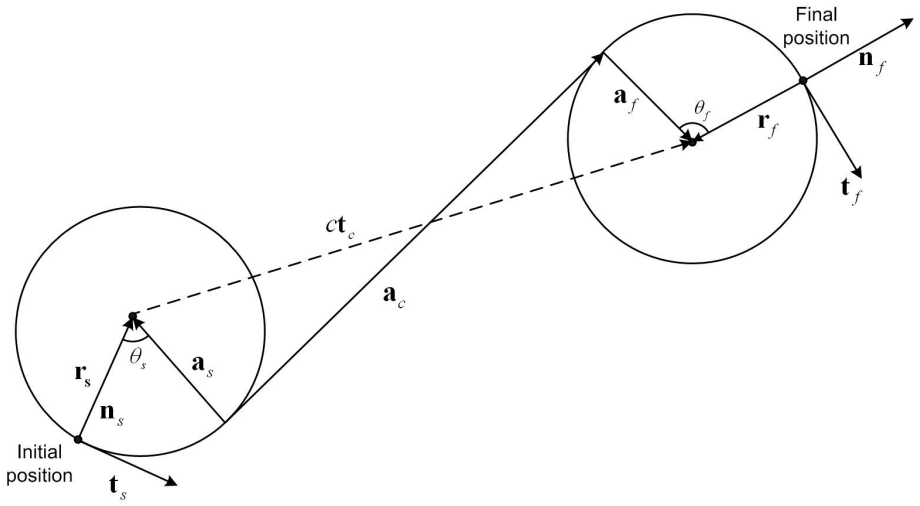
In this chapter, one of key enablers of the path planning scheme for cooperating UAVs is to use differential geometry concepts, developed in the previous study [55]. Path planning algorithms using differential geometry examine the evolution of guidance geometry over time to derive curvature satisfying UAV constraints. Guidance commands defining a manoeuvre profile can be then computed using the derived curvature of the guidance geometry. One of main advantages of this approach is that the number of design parameters can be significantly reduced whilst maintaining the guidance performance. Therefore, this approach will enable not only a fast design process and more lightweight algorithms, but will also generate safe and feasible paths for multiple UAVs. This is required for the integration of operational and physical constraints of the UAVs into the cooperative mission and path planning solution.

This chapter first presents the Dubins path planning which connects any two poses considering a bound on curvature as mentioned above. The Dubins path is considered because it is the shortest path, has simple geometry and is computationally efficient. The Dubins path is formed by concatenation of two circular arcs with their common tangent or by three consecutive tangential circular arcs, which are labelled as CSC or CCC path, respectively. Note that, CCC path is

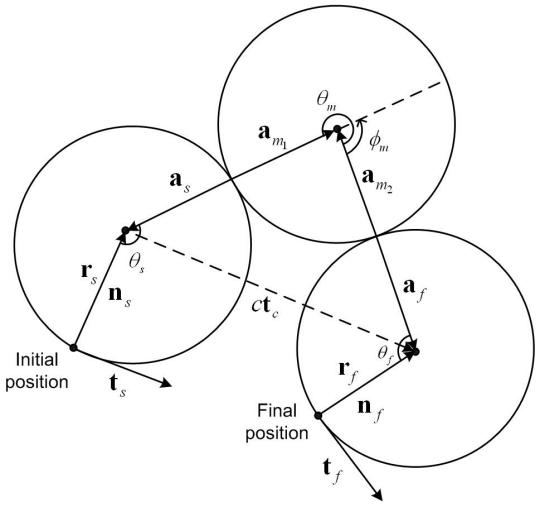
derived by using a principle of differential geometry which has not been done in the literature. Then, a new path planning technique is proposed using two constant curvature segments having more flexibility and fewer discontinuous points on a curvature command compared to the Dubins path, as well as satisfying the turning constraint. Fewer discontinuities in a guidance command are desirable for fixed-wing UAVs since they cannot exactly follow a discontinuous curvature (or bank angle) command (which occurs twice for the Dubins path and once for the proposed method); in reality, UAVs can track an approximated continuous command. Besides, Monte Carlo simulation results in Section 2.2.2 justify that the proposed method provides even less average curvature differences at those discontinuous points compared with the Dubins path at the expense of path length. In other words, the proposed approach provides a longer path length but with fewer discontinuity points as well as lower curvature differences than those of the Dubins path.

2.1 Dubins Path Planning Based on Differential Geometry

Motion in the plane is composed of either rectilinear or turning or angular motions. A straight line provides the shortest rectilinear motion and the circular arc provides the shortest turning or angular motion. The arc provides a constant turning radius which also satisfies the maximum curvature constraint, i.e. the minimum turning radius, which is a function of speed and maximum lateral acceleration. In this context, the Dubins path is formed either by concatenation of two circular arcs with their common tangent or by three consecutive tangential circular arcs. The former is CSC path, and the latter is CCC path, where 'C' stands for circular and 'S' stands for straight line segment. The CSC type path set D_{CSC} consists of $\{LSL, LSR, RSL, RSR\}$ and, the CCC path set D_{CCC} consists of $\{LRL, RLR\}$, where L stands for left turn and R for right turn. Geometry for both CSC and CCC type of the Dubins path is shown in Fig. 2.1. This section first derives the details of the construction of CCC paths using the principle of differential geometry to simplify the generation of the paths in a vector form. The vector sum for the position vector $\mathbf{p}$ which is a position of the final point $\mathbf{p}_f$ relative to the start point



(a) CSC path



(b) CCC path

Figure 2.1: Dubins path geometry

$\mathbf{p}_s$ in start axes is given by:

$$\begin{aligned}\mathbf{p} = \mathbf{p}_f - \mathbf{p}_s &= \mathbf{r}_s - \mathbf{a}_s + \mathbf{a}_{m_1} - \mathbf{a}_{m_2} + \mathbf{a}_f - \mathbf{r}_f \\ \mathbf{p} - \mathbf{r}_s + \mathbf{r}_f &= -\mathbf{a}_s + \mathbf{a}_{m_1} - \mathbf{a}_{m_2} + \mathbf{a}_f. \end{aligned} \tag{2.2}$$

The left hand side of this equation represents the vector $\mathbf{t}_c$ connecting the centres of the initial and the final circles. Hence,

$$c\mathbf{t}_c = -\mathbf{a}_s + \mathbf{a}_{m_1} - \mathbf{a}_{m_2} + \mathbf{a}_f \tag{2.3}$$

where c is the length of the centre vector. The remaining connecting vectors $\mathbf{a}_s$, $\mathbf{a}_{m_1}$, $\mathbf{a}_{m_2}$, and $\mathbf{a}_f$ can be expressed in terms of start basis vectors as:

$$\begin{aligned}
\mathbf{a}_s &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_s} \end{pmatrix} \\
\mathbf{a}_{m_1} &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_m} \end{pmatrix} \\
\mathbf{a}_{m_2} &= \mathbf{R}(\theta_s)' \mathbf{R}(\phi_m) \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_m} \end{pmatrix} \\
\mathbf{a}_f &= \mathbf{R}(\theta_s)' \mathbf{R}(\phi_m) \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_f} \end{pmatrix}
\end{aligned} \tag{2.4}$$

where

$$\phi_m = \pi - \cos^{-1}\left(\frac{c^2 - (\frac{1}{\kappa_s} + \frac{1}{\kappa_m})^2 - (\frac{1}{\kappa_m} + \frac{1}{\kappa_f})^2}{2(\frac{1}{\kappa_s} + \frac{1}{\kappa_m})(\frac{1}{\kappa_m} + \frac{1}{\kappa_f})} \right) \tag{2.5}$$

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{2.6}$$

and κ_s, κ_m and κ_f represent the curvature of the initial, the middle and final manoeuvre, respectively. Since a CCC type of the Dubins path consists *LRL* and *RLR*, the sign of k_s and k_f are the same, and that of k_s and k_m are different. Then, the center vector equation (2.3) now becomes

$$\mathbf{t}_c = \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} \pm(\frac{1}{\kappa_m} + \frac{1}{\kappa_f}) \sin(\phi_m) \\ \pm((\frac{1}{\kappa_s} + \frac{1}{\kappa_m}) + (\frac{1}{\kappa_m} + \frac{1}{\kappa_f}) \cos(\phi_m)) \end{pmatrix}. \tag{2.7}$$

This is a rotation equation; the right hand vector should have the unit magnitude, to give

$$\left| \frac{1}{c} \begin{pmatrix} \pm(1/\kappa_m + 1/\kappa_f) \sin(\phi_m) \\ \pm((1/\kappa_s + 1/\kappa_m) + (1/\kappa_m + 1/\kappa_f) \cos(\phi_m)) \end{pmatrix} \right| = 1. \tag{2.8}$$

This can be used to test for a feasible solution of the CCC path by:

$$1 - \frac{((1/\kappa_s + 1/\kappa_m) + (1/\kappa_m + 1/\kappa_f) \cos(\phi_m))^2}{c^2} > 0. \tag{2.9}$$

To compute the initial turning angle θ_s first, the equation can be written in the form

$$\mathbf{t}_c = \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \frac{1}{c} \begin{pmatrix} \beta & \gamma \\ \gamma & -\beta \end{pmatrix} \begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s \end{pmatrix} \tag{2.10}$$

where $\beta = \pm(1/\kappa_m + 1/\kappa_f)\sin(\phi_m)$ and $\gamma = \pm((1/\kappa_s + 1/\kappa_m) + (1/\kappa_m + 1/\kappa_f)\cos(\phi_m))$. Solving this for θ_s with $\beta^2 + \gamma^2 = c^2$ from Eq. (2.8) gives:

$$\begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \frac{1}{c}\begin{pmatrix} \beta & \gamma \\ \gamma & -\beta \end{pmatrix}\mathbf{t}_c. \tag{2.11}$$

The final turning angle θ_f can be determined by using

$$\begin{aligned} \theta_d &= \theta_s - \theta_m + \theta_f \\ \theta_f &= \theta_d - \theta s + \theta_m \end{aligned} \tag{2.12}$$

where θ_d represents the difference between the initial and final heading angle and $\theta_m = \pm(\pi + |\phi_m|)$. Finally, the path length is computed by summation of the arc lengths as:

$$L_{CCC} = \frac{\theta_s}{\kappa_s} + \frac{\theta_m}{\kappa_m} + \frac{\theta_f}{\kappa_f}. \tag{2.13}$$

Similarly, construction of CSC paths is explained following from [58]. First of all, the vector sum for the position vector $\mathbf{p}$ is given by:

$$\mathbf{p} = \mathbf{r}_s - \mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f - \mathbf{r}_f. \tag{2.14}$$

Then, the vector $\mathbf{t}_c$ becomes:

$$c\mathbf{t}_c = \mathbf{p} - \mathbf{r}_s + \mathbf{r}_f = -\mathbf{a}_s + \mathbf{a}_{m_1} - \mathbf{a}_{m_2} + \mathbf{a}_f. \tag{2.15}$$

The remaining connecting vectors $\mathbf{a}_s$, $\mathbf{a}_c$, and $\mathbf{a}_f$ can be expressed in terms of start basis vectors as:

$$\begin{aligned} \mathbf{a}_s &= \mathbf{R}(\theta_s)'\begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_s} \end{pmatrix} \\ \mathbf{a}_c &= \mathbf{R}(\theta_s)'\begin{pmatrix} a \\ 0 \end{pmatrix} \\ \mathbf{a}_f &= \mathbf{R}(\theta_s)'\begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_f} \end{pmatrix}. \end{aligned} \tag{2.16}$$

Using above equations, the centre vector equation (2.15) can be expressed as:

$$\mathbf{t}_c = \mathbf{R}(\theta_s)'\frac{1}{c}\begin{pmatrix} a \\ \pm\frac{1}{\kappa_f} - \pm\frac{1}{\kappa_s}) \end{pmatrix}. \tag{2.17}$$

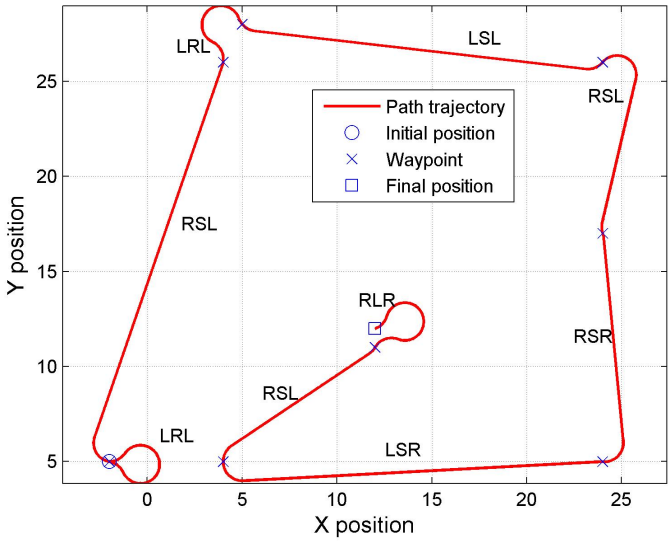


Figure 2.2: Example Dubins paths ($\kappa = 1$)

To compute the initial turning angle θ_s, the equation can be written as:

$$\mathbf{t}_c = \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} \sqrt{c^2 - \left[(\pm 1/\kappa_f) - (\pm 1/\kappa_s)\right]^2} \\ (\pm 1/\kappa_f) - (\pm 1/\kappa_s) \end{pmatrix}. \tag{2.18}$$

Solving this with the same way as CCC path case gives the initial turning angle θ_s, and the final angle θ_f can then be determined by using

$$\begin{aligned} \theta_d &= \theta_s + \theta_f \\ \theta_f &= \theta_d - \theta s. \end{aligned} \tag{2.19}$$

The path length for this CSC case is computed by summation of the arc lengths and connecting tangent length as:

$$L_{CSC} = \frac{\theta_s}{\kappa_s} + a + \frac{\theta_f}{\kappa_f}. \tag{2.20}$$

If both of the initial and final heading angle are given, a set of six paths including CSC and CCC path $\{LSL, LSR, RSL, RSR, RLR, LRL\}$ can be produced, and the shortest path is selected from these set of available paths. Figure 2.2 shows the example of Dubins paths including both CCC and CLC type paths where the initial and final path curvature is one, and each way-point has distinctive approach angle defined.

Following [59], now let us briefly look at safe paths. The Dubins paths satisfy the maximum bound on curvature. However, safety of UAVs needs additional constraints to be met to produce safe flyable paths. In this respect, the Dubins paths have to meet the safety conditions: (i) minimum separation distance and (ii) non-intersection of paths at equal lengths. The minimum separation distance d_{sep} between any two UAVs should at least be equal to the summation of corresponding radii of the safety circles as given:

$$d_{sep} \geq R_{s,1} + R_{s,2} \tag{2.21}$$

where $R_{s,i}$ represents the radii of the safety circle of the UAV i. Although two Dubins paths failed to meet the minimum separation distance, this does not necessarily imply a collision. To verify a collision, the path length L_{int} of each path from its starting point to the point of failure on minimum separation should be calculated. The difference between the path lengths $d_{int} = |L_{int,1} - L_{int,2}|$ must be less than the summation of corresponding radii of safety circles to confirm a potential collision as given:

$$d_{int} \geq R_{s,1} + R_{s,2} \tag{2.22}$$

In the event of failure of both conditions, the replanning should be done by increasing the radius of the turning circles or choosing the next shortest path from the Dubins set. For a group of N UAVs, taking r UAVs at a time, the safety conditions have to be tested for n_u times, where

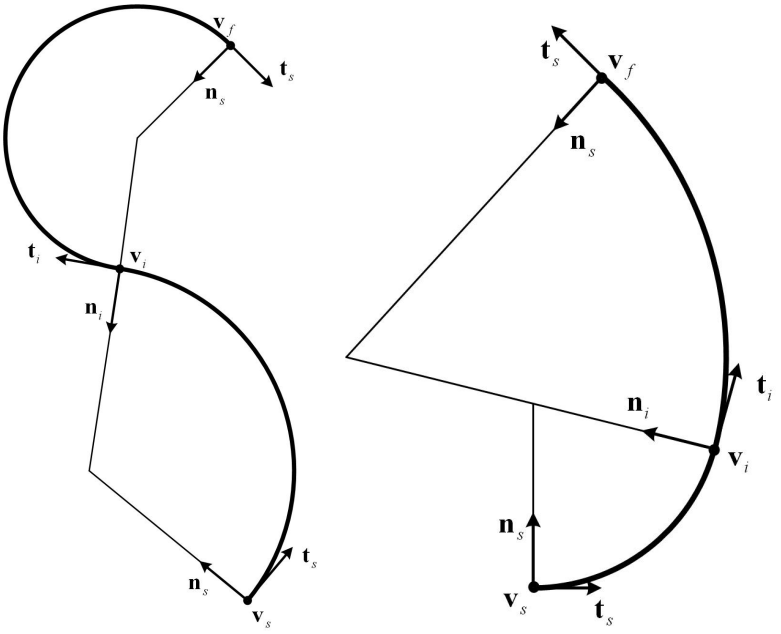
$$n_u = 2\frac{N!}{r!(N-r)!} \tag{2.23}$$

This Dubins path planning will be applied to coordinated road-network search route planning algorithms in Chapter 3.

2.2 Path Planning Using Constant Curvature Segments

The path planning technique using constant curvature segments is to connect two configurations by using two arc segments of the constant curvature along with the consideration of a bound on the curvature. A single arc segment between two

positions only has one degree of freedom: an arc curvature. This is not enough to be able to match the pose constraint at both positions since at least two degrees of freedom are necessary. Extra degrees of freedom are thus required to ensure C^1 contact at initial and final positions, which represents that the first derivative is continuous at both points. One solution to increase the degrees of freedom is to introduce an intermediate vertex such that the line segment is replaced by two arc segments of different curvature as shown in Fig. 2.3 [10]. Note that this solution is not unique nor necessarily time optimal. The initial and final position are identified as $\mathbf{v}_s$ and $\mathbf{v}_f$ with the intermediate vertex given by $\mathbf{v}_i$. Hence, two arcs of different curvature will connect two configurations via intermediate vertex with C^1 contact. Two orientations for the intermediate arcs are possible as shown in Fig. 2.3. Alternatively, the paths can be divided into four cases based on the



(a) Orientation 1 *(b) Orientation 2*

Figure 2.3: Arc segments with C^1 contact and intermediate vertex

orientation of initial and final configurations as:

1. First case: initial and final orientations are towards $\mathbf{p}_t$,
2. Second case: initial and final orientations are opposite to $\mathbf{p}_t$,
3. Third case: initial orientation is towards $\mathbf{p}_t$ while final is opposite to it,
4. Fourth case: initial orientation is opposite to $\mathbf{p}_t$ while final is towards it,

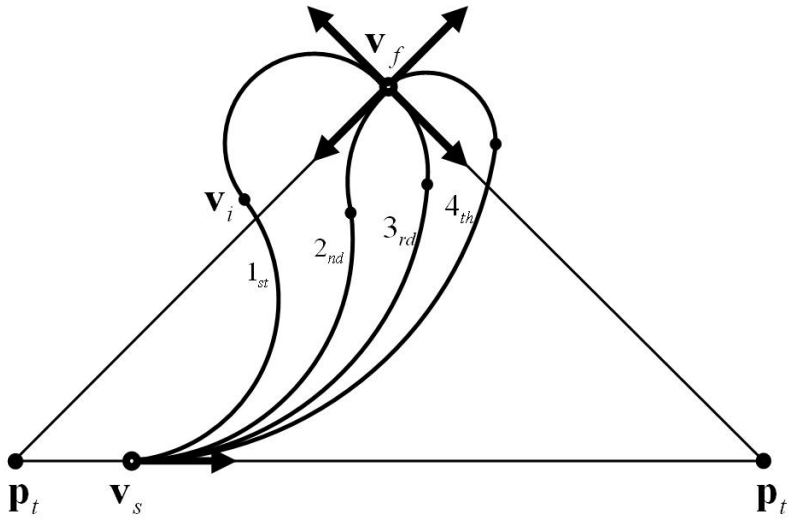


Figure 2.4: Four possible cases of the paths using two arc segments

where $\mathbf{p}_t$ represents a tangent intersection point of two orientations as shown in Fig. 2.4. Since the conditions for some of the cases are either identical or similar, only the first and the third case will be explained in detail from the following.

2.2.1 First and Second Case

The first case is defined as the one with initial and final orientations towards their tangent intersection point $\mathbf{p}_t$ as shown in Fig. 2.5. Using the angles inside the quadrilateral formed by points $\mathbf{p}_t, \mathbf{v}_s, \mathbf{v}_f$ and $\mathbf{v}_i$ gives angle θ as:

$$\theta = 2\pi - \phi_s - \phi_f - \psi \tag{2.24}$$

where ϕ_s and ϕ_f represent the angles between the tangents and the chord c connecting initial and final positions to the intermediate vertex, respectively. Considering the triangle $(\mathbf{v}_s, \mathbf{v}_i, \mathbf{o}_s)$ and $(\mathbf{v}_f, \mathbf{v}_i, \mathbf{o}_f)$, angle θ can also be expressed as:

$$\theta = \pi - \zeta_s - \zeta_f \tag{2.25}$$

where $\zeta_s = \frac{\pi}{2} - \phi_s$ and $\zeta_f = \frac{\pi}{2} - \phi_f$. Combing Eq. (2.24) and Eq. (2.25) gives:

$$\phi_f = \pi - \frac{\psi}{2} - \phi_s. \tag{2.26}$$

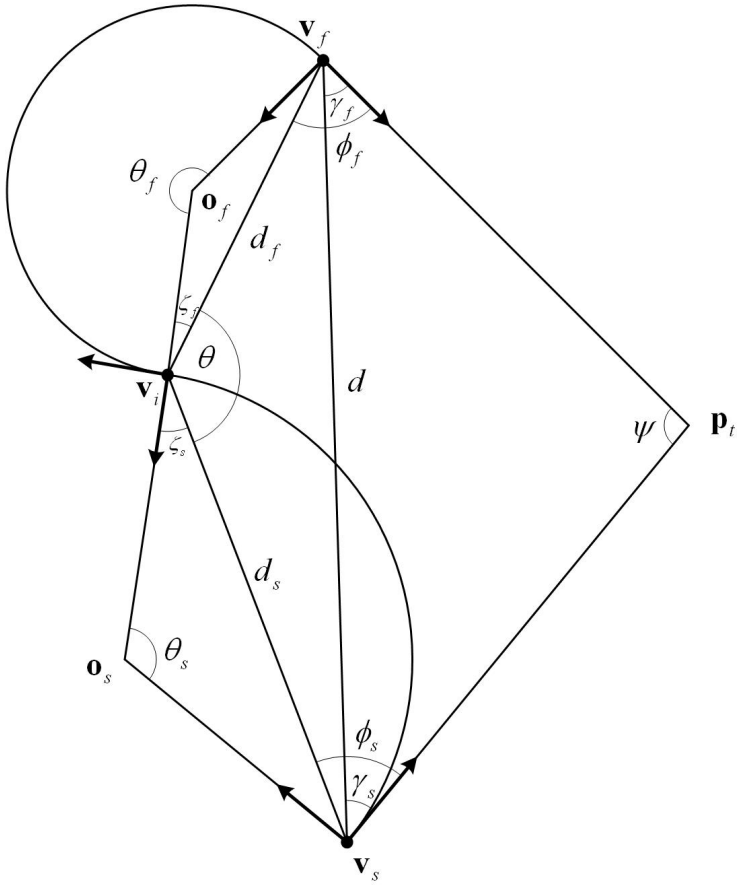


Figure 2.5: Geometry for the first case

Both $\phi_s \geq 0$ and $\phi_f \geq 0$ are required to maintain consistent arc direction by orienting the directed tangent vector at the intermediate vertex, which gives a boundary for ϕ_s as:

$$0 \leq \phi_s \leq \pi - \frac{\psi}{2}. \tag{2.27}$$

Another limitation on the angle comes from the first case requirement itself, as given:

$$-\pi < \theta_s - \theta_f \leq 0 \tag{2.28}$$

where angular values are subtracted since the direction of the turn manoeuvre applied in this case is opposite to each other. As shown in Fig. 2.6, given θ_s, the dashed line where $\theta_s - \theta_f < 0$ represents the first case with initial and final orientations towards their tangent intersection, while the solid line represents the

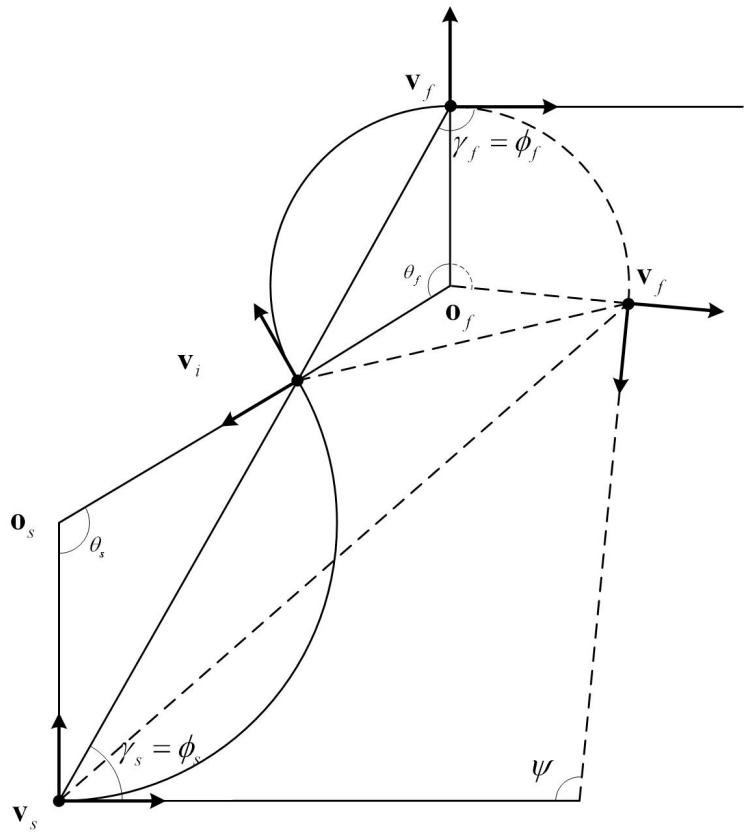


Figure 2.6: Feasible and limit conditions for the first case

limit condition for this case, where $\theta_s - \theta_f = 0$, $\phi_s = \gamma_s$, and $\phi_f = \gamma_f$. This relation gives the following inequality:

$$\phi_s \geq \gamma_s \tag{2.29}$$

$$\phi_f \geq \gamma_f \tag{2.30}$$

where γ_s and γ_f represent angles between tangents and the chord segment. Using Eq. (2.26) and Eq. (2.29), the final boundary of ϕ_s is obtained:

$$\gamma_s \leq \phi_s \leq \pi - \frac{\psi}{2} - \gamma_f. \tag{2.31}$$

To obtain other parameters for the generation of the path, the arc chord length d_s and d_f are first obtained using triangle $(\mathbf{v}_s, \mathbf{v}_i, \mathbf{v}_f)$ and sine rule as:

$$\begin{aligned} d_s &= d\frac{\sin(\phi_f - \gamma_f)}{\sin(\theta)} \\ d_f &= d\frac{\sin(\phi_s - \gamma_s)}{\sin(\theta)} \end{aligned} \tag{2.32}$$

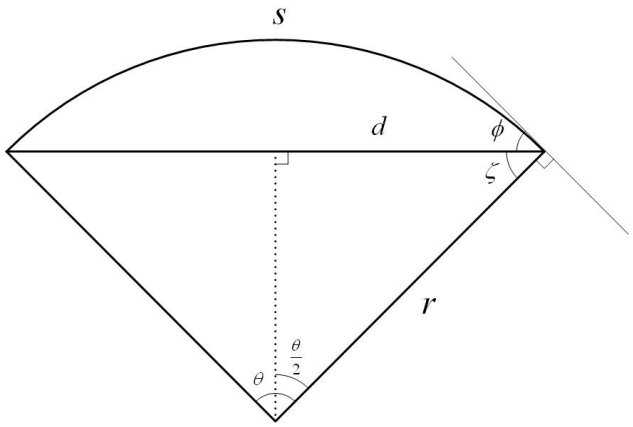


Figure 2.7: Geometric relation for the curvature

where d represents the vertex chord length between the vertices $\mathbf{v}_s$ and $\mathbf{v}_f$. For a generic circle arc as shown in Fig. 2.7, the curvature κ is obtained by following geometric relations:

$$d = \frac{\sin\left(\frac{\theta}{2}\right)}{\left(\frac{\theta}{2}\right)} s \tag{2.33}$$

$$\kappa = \frac{\theta}{s} = \theta \frac{\sin\left(\frac{\theta}{2}\right)}{\frac{\theta}{2} d} = \frac{2\sin(\phi)}{d}. \tag{2.34}$$

Using the same way as above, the curvature of each segment κ_s and κ_f can then be determined as:

$$\begin{aligned} \kappa_s &= \frac{2\sin(\phi_s)}{d_s} \\ \kappa_f &= \frac{2\sin(\phi_f)}{d_f}. \end{aligned} \tag{2.35}$$

The path planning technique explained above produces the path that consists of two arcs with different curvature values κ_s and κ_f. However, in order to produce flyable paths for the UAV, physical constraints of the UAV flight, the limited curvature or the minimum turning radius, should be taken into account. In the following, these constraints are dealt with by changing the boundary on ϕ_s, that is the only degree of freedom available, satisfying the bound on the curvature κ_{max} or turn radius r_m.

First of all, let us check the characteristic of the curvature or turning radius for initial arc depending on the value of ϕ_s using Eq. (2.26), Eq. (2.32) and Eq. (2.35)

as:

$$r_s = \frac{1}{\kappa_s} = d\frac{\sin(\phi_f - \gamma_f)}{2\sin\theta\sin\phi_s} = d\frac{\sin(\alpha_{s_1} - \phi_s)}{2\sin\theta\sin\phi_s} \tag{2.36}$$

where $\alpha_{s_1} = \frac{\psi}{2} + \gamma_s$. The characteristic of the turning radius for initial arc is obtained as the following lemma.

Lemma 2.2.1. *(Characteristic of the turning radius for the initial arc of 1st case): For $0 < \psi < \pi$, $d > 0$ and $0 \leq \phi_s < \pi$, initial turning radius r_s is monotonically decreasing varying ϕ_s from its minimum to maximum bound in the first case.*

Proof. Considering the boundary of ϕ_s from Eq. (2.31), r_s for the minimum and the maximum bound of ϕ_s can be obtained as:

$$r_{s,\phi_{s_{sup}}} = d\frac{\sin(\phi_f - \gamma_f)}{2\sin\theta\sin\phi_s} = d\frac{\sin\left(\pi - \frac{\psi}{2} - \phi_{s_{sup}} - \gamma_f\right)}{2\sin\theta\sin\phi_{s_{sup}}} = 0 \tag{2.37}$$

$$\begin{aligned} r_{s,\phi_{s_{inf}}} &= d\frac{\sin\left(\pi - \frac{\psi}{2} - \phi_{s_{inf}} - \gamma_f\right)}{2\sin\theta\sin\phi_{s_{inf}}} = d\frac{\sin\left(\pi - \frac{\psi}{2} - \gamma_s - \gamma_f\right)}{2\sin\theta\sin\gamma_s} \\ &= d\frac{\sin\frac{\psi}{2}}{2\sin\theta\sin\gamma_s} = d\frac{1}{2\sin\gamma_s} \end{aligned} \tag{2.38}$$

where $\phi_{s_{sup}} = \pi - \frac{\psi}{2} - \gamma_f$, $\phi_{s_{inf}} = \gamma_s$, and $\sin\theta = \sin\frac{\psi}{2}$ since

$$\sin\theta = \sin(2\pi - \phi_s - \phi_f - \psi) = \sin\frac{\psi}{2}.$$

Moreover, differentiating Eq. (2.36) with respect to ϕ_s gives:

$$\frac{\partial r_s}{\partial\phi_s} = \frac{d}{2\sin\theta}\frac{-\cos(\alpha_{s_1} - \phi_s)\sin\phi_s - \cos\phi_s\sin(\alpha_{s_1} - \phi_s)}{\sin^2\phi_s}. \tag{2.39}$$

It can be shown that Eq. (2.39) is always negative for $0 < \psi < \pi$, $d > 0$ and $0 \leq \phi_s < \pi$ using simple trigonometry. Negative values for derivative, together with r_s values for the minimum and the maximum bound of ϕ_s result in monotonically decreasing r_s with increasing ϕ_s. □

Having analysed the characteristic of the initial turning radius, a new limit on ϕ_s satisfying the bound on the turning radius r_m can be obtained by imposing the following condition:

$$r_s = d\frac{\sin(\phi_f - \gamma_f)}{2\sin\theta\sin\phi_s} \geq r_m. \tag{2.40}$$

Using Eq. (2.26) and trigonometry identities with Eq. (2.40) gives:

$$\begin{aligned} \frac{\sin\left(\frac{\psi}{2}+\phi_s+\gamma_f\right)}{\sin\theta\sin\phi_s} &\geq \frac{2r_m}{d} \\ \cos\left(\frac{\psi}{2}+\gamma_f\right)+\cot\phi_s\sin\left(\frac{\psi}{2}+\gamma_f\right) &\geq \frac{2r_m\sin\frac{\psi}{2}}{d} \\ \cot\phi_s &\geq \frac{2r_m\sin\frac{\psi}{2}}{d\sin\left(\frac{\psi}{2}+\gamma_f\right)}-\cot\left(\frac{\psi}{2}+\gamma_f\right). \end{aligned} \tag{2.41}$$

The feasible upper bound on ϕ_s to meet the turning radius constraint can be obtained as:

$$\phi_s \leq \tan^{-1}\left[\frac{1}{\beta_{s_1}-\cot\left(\frac{\psi}{2}+\gamma_f\right)}\right] \tag{2.42}$$

where

$$\beta_{s_1}=\frac{2r_m\sin\frac{\psi}{2}}{d\sin\left(\frac{\psi}{2}+\gamma_f\right)}.$$

The second part of the path characterised by the curvature κ_f can be analysed in the same manner as above: the turning radius of final arc r_f tends to be a finite value for $\phi_{s_{sup}}$, while it tends to be zero for $\phi_{s_{inf}}$ as opposed to the previous initial arc case. By expressing r_f as a function of ϕ_f gives:

$$r_f = d\frac{\sin(\phi_s-\gamma_s)}{2\sin\theta\sin\phi_f} = d\frac{\sin(\alpha_{f_1}-\phi_f)}{2\sin\theta\sin\phi_f} \tag{2.43}$$

where $\alpha_{f_1}=\frac{\psi}{2}+\gamma_f$. Then, the characteristic of the turning radius for final arc is obtained as the following lemma.

Lemma 2.2.2. *(Characteristic of the turning radius for the final arc of 1st case): For $0<\psi<\pi$, $d>0$, $0\leq\phi_s<\pi$ and $0\leq\phi_f<\pi$, final turning radius r_f is monotonically decreasing with increasing ϕ_f, but monotonically increasing with increasing ϕ_s in the first case.*

Proof. It can be shown that r_f is monotonically decreasing varying ϕ_f from its minimum to maximum bound using the same steps of the initial arc case. However, recalling that $\phi_f=\pi-\frac{\psi}{2}-\phi_s$, it is clear that ϕ_s increases as ϕ_f decreases, resulting in monotonically increasing r_f with increasing ϕ_s. □

To ensure the feasibility of the final arc solution, the following condition is imposed:

$$r_f = d\frac{\sin(\phi_s - \gamma_s)}{2\sin\theta\sin\phi_f} \geq r_m. \tag{2.44}$$

The feasible upper bound on ϕ_f to meet the turning radius constraint can be obtained as:

$$\phi_f \leq \tan^{-1}\left[\frac{1}{\beta_{f_1} - \cot\left(\frac{\psi}{2} + \gamma_s\right)}\right] \tag{2.45}$$

where

$$\beta_{f_1} = \frac{2r_m \sin\frac{\psi}{2}}{d\sin\left(\frac{\psi}{2} + \gamma_s\right)}.$$

The feasible lower bound on ϕ_s can be obtained as:

$$\begin{aligned} \phi_f = \pi - \frac{\psi}{2} - \phi_s &\leq \tan^{-1}\left[\frac{1}{\beta_{f_1} - \cot\left(\frac{\psi}{2} + \gamma_s\right)}\right] \\ \phi_s &\geq -\tan^{-1}\left[\frac{1}{\beta_{f_1} - \cot\left(\frac{\psi}{2} + \gamma_s\right)}\right] - \frac{\psi}{2} + \pi. \end{aligned} \tag{2.46}$$

By choosing appropriate ϕ_s using the new boundary from Eq. (2.42) and Eq. (2.46) and following ϕ_f from Eq. (2.26), the flyable path for the UAV is generated.

The second case deals with the situation where both initial and final orientations are opposite to their tangent intersection point. An example of such a configuration is given in Fig. 2.8. A feasible solution requires:

$$0 < \theta_s - \theta_f \leq \pi. \tag{2.47}$$

In this case, the geometrical configuration of triangle ($\mathbf{v}_s$, $\mathbf{v}_i$, $\mathbf{v}_f$) and ($\mathbf{v}_s$, $\mathbf{p}_t$, $\mathbf{v}_f$) does not change compared to the first case since initial and final positions are the same but with opposite orientations, and the only difference is rotation angles θ_s and θ_f. By considering the fact that these angles do not affect the process of computing the parameters for the arc curvature, it can be concluded that the path constraints for the physical feasibility are obtained in a similar manner as for the first case. The details of the third and fourth cases can be found in Appendix A.

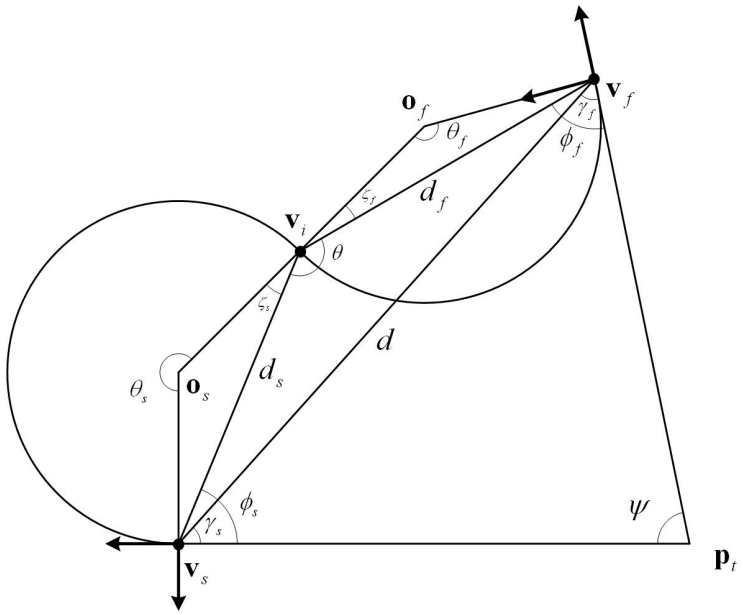


Figure 2.8: Geometry for the second case

To meet C^1 contact at both positions and turning constraints, ϕ_s is selected appropriately within its boundary depending on the cost function such as the path length or any other metric as shown in Algorithm 1, which represents how to implement the proposed path planning through pseudo-code.

Algorithm 1 Path planning using two constant curvature segments

Input: $P_s(x_s, y_s, \sigma_s)$, $P_f(x_f, y_f, \sigma_f)$ and κ_{max} (or r_m)

Output: r(t) = path(θ_{s,i^*}, θ_{f,i^*}, κ_{s,i^*}, κ_{f,i^*})

1: Select the case amongst 1$\sim$ 4_{th} cases
2: Compute ψ, γ_s and γ_f
3: Find $\phi_{s,min}$ and $\phi_{s,max}$
4: **for** $\phi_{s,i} = \phi_{s,min}$ to $\phi_{s,max}$ **do**
5: Compute $\zeta_{s,i}$, $\zeta_{f,i}$, $\theta_{s,i}$, $\theta_{f,i}$, $d_{s,i}$, $d_{f,i}$, $\kappa_{s,i}$, $\kappa_{f,i}$
6: $L_i = |\frac{\theta_{s,i}}{\kappa_{s,i}}| + |\frac{\theta_{f,i}}{\kappa_{f,i}}|$ {L_i := path length}
7: $\phi_{s,i} \leftarrow \phi_{s,i} + \triangle\phi_s$
8: **end for**
9: Find $i^* = \arg\min_i L_i$

Given the boundary conditions and turning constraints, reachable space can be investigated by expressing the chord length d from the triangle $(\mathbf{v}_s, \mathbf{v}_i, \mathbf{v}_f)$ in

Fig. 2.8 as:

$$\begin{aligned} d &= \sqrt{{d_s}^2 + {d_f}^2 - 2d_s d_f \cos\theta} \\ &= \sqrt{4{r_s}^2(1-\cos\theta_s) + 4{r_f}^2(1-\cos\theta_f) - 8r_s r_f(1-\cos\theta_s)(1-\cos\theta_f)\cos\theta}. \end{aligned} \tag{2.48}$$

Then, the minimum value of d for a feasible path with the minimum turning radius r_m for all feasible θ_s and θ_f is given as:

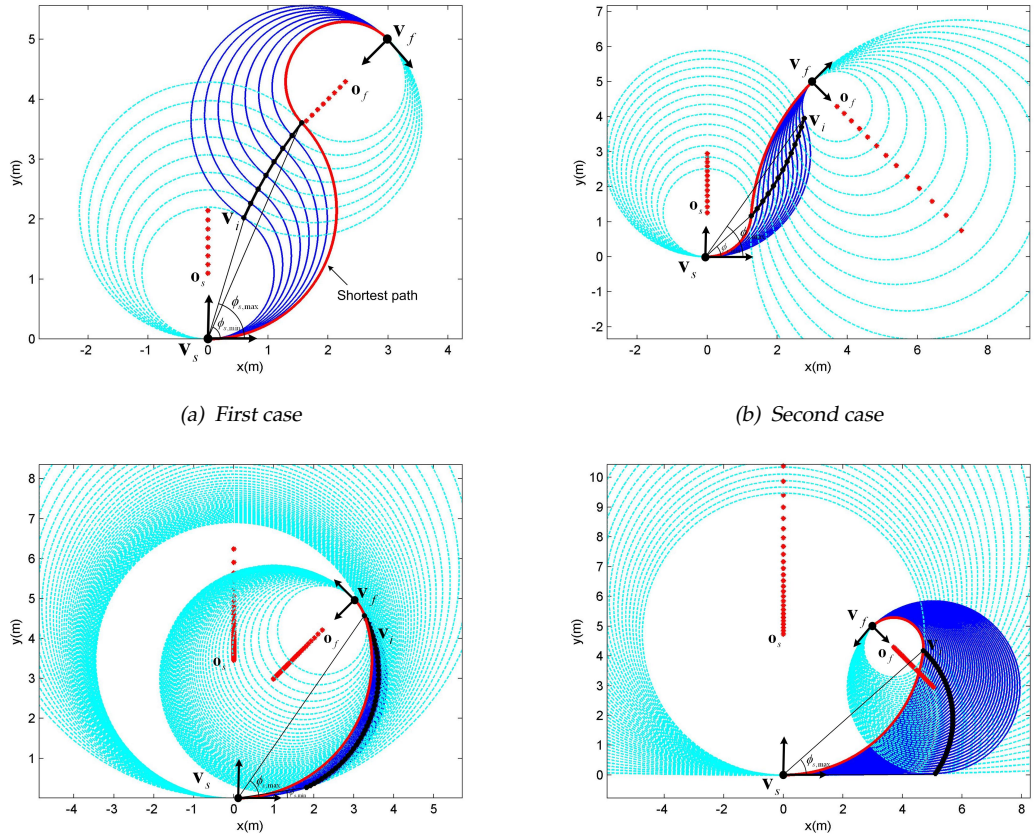
$$d_{min} = \begin{cases} 4r_m\sqrt{1-2\cos\left(\pi-\frac{\psi}{2}\right)}, & \text{if case 1 or 2} \\ 2r_m, & \text{otherwise.} \end{cases} \tag{2.49}$$

Thus, two configurations whose Euclidean distance between is greater than d_{min} satisfy one of conditions for having a solution. However, note that this is a necessary condition since there are other boundary conditions to be satisfied as defined through this section. Besides, configurations which make linear paths need to be handled as exceptions as they require infinite turning radius or zero curvature.

2.2.2 Numerical Simulations

To verify the feasibility and benefits of the proposed approach, the path planning using constant curvature segments is simulated using initial configuration (x, y, σ_s) = (0, 0, 0) and final configuration = (3, 5, σ_f = ($-\frac{\pi}{4}$, $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$)) with bounds on curvature of $\kappa_s = 1$ and $\kappa_f = 1$. Figure 2.9 shows the examples of the paths using the solution ϕ_s minimizing the path length. This figure also shows the locus of all feasible intermediate vertex $\mathbf{v}_i$ with the corresponding possible paths. In Fig. 2.9(a), as ϕ_s increases, initial turning radius decreases and final turning radius increases confirming Lemma 2.2.1 and 2.2.2 for the first case of path solutions. Similarly, as ϕ_s increases, both initial and final turning radius decreases for the third and fourth case as shown in Fig. 2.9(c)~(d).

Figure 2.10 compares the average path length and curvature difference at intermediate vertex (or discontinuous points when the segment changes from arc to straight line for the Dubins path) between the proposed method and Dubins



(a) First case

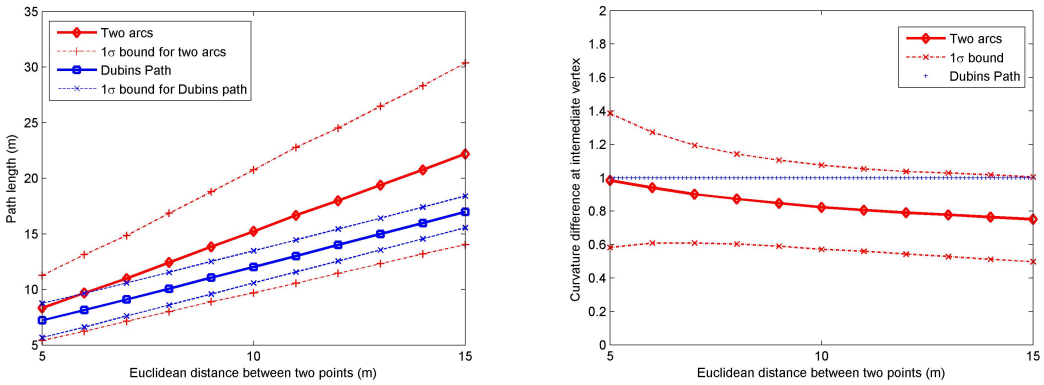
(b) Second case

(c) Third case

(d) Fourth case

Figure 2.9: Examples of path using constant curvature segments ($(\kappa_s, \kappa_f) = (1, 1)$, *blue lines are possible paths, and red dots represent the trace of the center of each curvature segment according to* ϕ_s.)

path. Simulations are performed a hundred times for each configuration set by changing final configuration $(E\cos\theta_m, E\sin\theta_m, \sigma_{f,rand})$ for all $5 \leq E \leq 15$ and $0 \leq \theta_m < 2\pi$ with fixed initial configuration $(0, 0, 0)$ and a bound on curvature of $[-1, 1]$, where $\sigma_{f,rand}$ represents a randomly generated angle. As a Euclidean distance E between two points increases, the average path length of the proposed method gets longer than that of the Dubins path, and curvature difference decreases whereas that of Dubins remains the same. Consequently, it can be conjectured that the proposed algorithm provides not only fewer discontinuous points (which is one for the proposed method and two for the Dubins path), but also less discontinuity (i.e. curvature difference) since as can be seen for case 3 and 4, curvature differences are much smaller than those of the Dubins paths at the expense of path length.



(a) Path length *(b) Curvature difference at intermediate vertex*

Figure 2.10: Comparison between the proposed method and Dubins path

The proposed path planning algorithm satisfies the kinematic constraint of the UAV with only one discontinuity at the intermediate vertex. It can also shape the path (or equivalently, change the path length) using one parameter ϕ_s with its boundary condition according to the initial and final configurations. Thus, this technique is applied to coordinated standoff target tracking, which makes two UAVs arrive at a desired orbit and angular separation in Section 6.1.

Chapter 3

Coordinated Road-Network Search Route Planning

3.1 Introduction

For surveillance, inspection, and intelligence missions, UAVs need to patrol some region and gather information. These missions often constrain UAVs' path and trajectory. For instance, if UAVs are utilized to get some information of enemy's activities on the specific roads and military bases, or to observe the traffic of ports or roads, they should fly over only those roads or region rather than patrol whole terrain. In this chapter, this problem is defined as the road search route planning problem, which can be related to a vehicle routing problem: how multiple UAVs can patrol or cover a specific region of interest having a complex road-network within space, dynamic, and operational constraints.

The vehicle routing problem has been mainly handled in the operational research area [60] and can be generally classified by two categories: one is the Traveling Salesman Problem (TSP) which finds a shortest circular trip through a given number of cities, and the other is the Chinese Postman Problem (CPP) finding the shortest path with considering path constraints on an entire network of road. The TSP using multiple UAVs can be considered as a task assignment problem to minimise the cost of time or energy for a certain mission by assigning each target to UAVs, for which binary linear programming [61], iterative network flow [62] and tabu search algorithm [63] have been proposed. Choset [64] surveys research results in coverage path planning that determines a path for a robot to pass over all points in its free space. The CPP and its variants [65] are normally used for ground vehicle applications such as road maintenance, snow disposal [66], boundary coverage [67], and graph searching and sweeping [68].

In the aforementioned works, since the general vehicle routing algorithms approximate their path to a straight line shape to reduce computational load, the physical constraints imposed on the vehicle are not to be addressed. Re-

cently, Dubins TSP (DTSP) algorithms were developed which can accommodate the physical constraints using the so-called Dubins nonholonomic planar vehicle model constrained to move along paths of bounded curvature without reversing direction. For a single vehicle, Salva *et al.* [69] proposed an alternating DTSP algorithm based on the solution to the conventional TSP and on an alternating heuristic to assign the target orientation at each target point. Ny *et al.* [70] also developed a DTSP algorithm using a heading discretisation. The similar problem but using multiple agents is considered in [37, 71, 72]. However, these physical constraints have been rarely dealt with in the context of the CPP.

This chapter presents the road-network search route planning algorithms by which multiple airborne platforms are able to efficiently patrol every road segment identified in the map in the context of the CPP. The first part of this chapter introduces a conventional CPP algorithm for the case that ground vehicles moves along a connected road-network. Following this, the CPP algorithm is modified to consider a general type of roadmap including unconnected roads as well as operational and physical constraints on speed and minimum turning radius of UAVs. Previous work [73] regarding the modified CPP (mCPP) for the road-network search route planning was to formulate Multi-choice Multi-dimensional Knapsack Problem (MMKP) so as to find an optimal solution minimising a path length or a flight time and to solve it via mixed integer linear programming (MILP).

The main contributions of this chapter are threefold. Firstly, to overcome the computational burden of the MMKP algorithm and to induce a real time solution, a nearest insertion algorithm combining with an auction-based negotiation is newly proposed to be applied for the search route planning by multiple UAVs. Secondly, in order to accommodate the physical constraints of the UAV in the search pattern design, this study uses the Dubins trajectory [44] which is the shortest path connecting two configurations represented under the constraints of a bound on curvature as described in Section 2.1. Although a CSC (circular-straight-circular) type of the Dubins path has been generally used for the path planning of autonomous systems, this study exploits both a CCC (circular-circular-circular) and a CSC type of the Dubins path to precisely cover a densely distributed road environment. Lastly, this study systematically investigates the performance of the proposed algorithm depending on different map sizes, path planning methods

(straight line and Dubins path) and the number of UAVs by using Monte Carlo simulations. Based on these results, an efficient UAV team size and path planning method is advised for the specific road-network search mission considered in this chapter. Furthermore, to clarify the benefit of the proposed algorithm, this study compares the performance of the MMKP optimisation and the approximation algorithm in terms of computational load and flight time for a specific road-network search scenario.

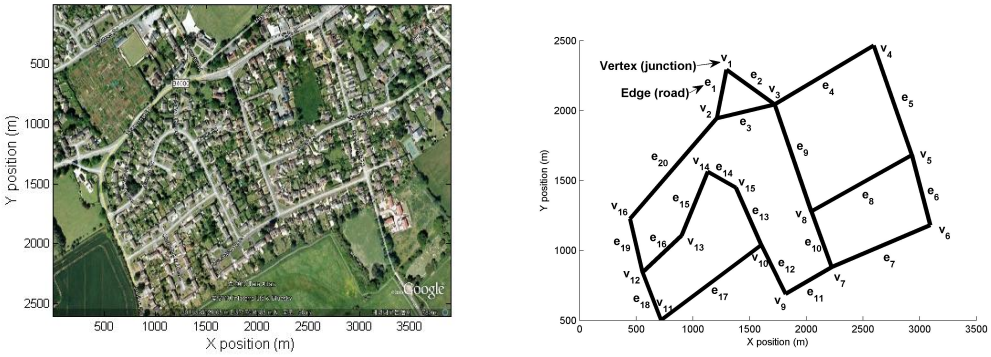
The remainder of this chapter is organised as follows. Section 3.2 defines the problem of road-network search route planning introducing the TSP and the CPP. Then, Section 3.3 proposes the road-network search route planning algorithms for multiple UAVs based on the MILP optimisation and the approximation algorithm using nearest insertion and auction negotiation. Lastly, the performances and the properties of the proposed algorithm are verified via numerical simulations in Section 3.4. Summary of this work is given in Section 3.5.

3.2 Road-Network Search Route Planning

For a search mission, a road-network is established as a set of straight line joining waypoints. These waypoints are located either on road junctions or along the roads with sufficient separations between them to allow the accurate representation of the curved road by using a set of straight lines. This study selects a sample road-network as shown in Fig. 3.1(a), which is based on the Google map of some village in the UK. This road-network can be translated to a graph composed of straight line segments as shown in Fig. 3.1(b). In order to search all the roads within the map of interest, there are generally two typical routing problems [74].

Traveling Salesman Problem (TSP): A salesman has to visit several cities (or road junctions). Starting at a certain city, the TSP is to find a route of minimum travel distance on which the salesman traverses each of the destination cities exactly only once (and for the closed TSP, leads him back to his starting point).

Chinese Postman Problem (CPP): A postman has to deliver mail for a network of streets. Starting at a given point, e.g. the post office, the CPP is to find a route of minimum travel distance allowing the postman to stop by each street at least once (and for the closed CPP, leading him back to the post office).



(a) The satellite map of a village in the UK *(b) Its graphic representation by straight line segments*

Figure 3.1: The graphic representation of a road-network

This study considers the CPP and its variants, which involve constructing a tour of all the roads with the shortest distance of the road-network. Typically, the road-network is mapped to an undirected graph $G = (V, E)$, having edge weights $w : E \rightarrow \mathbb{R}_0^+$, where the roads are represented by the edge set $E = \{e_1, \ldots, e_n\}$, and the road junctions are represented by the vertex set $V = \{v_1, \ldots, v_m\}$ as numbered in Fig. 3.1(b). Each edge $e_i = \{v_{e_{i,1}}, v_{e_{i,2}}\}$ is weighted with its length or the amount of time required to traverse it. The basic CPP algorithm involves first constructing an even graph from the road-network which has a set of vertices with an even number of edges attached to them producing a pair of entry and exit points. When the road-network graph has junctions with an odd number of edges, some roads therefore need to be selected for multiple visits as exceptions by the postman to make the even graph. The search pattern (tour) of the even graph is calculated by determining the Euler tour of the graph [60], which visits every edge of the even graph exactly once, or twice for duplicated ones.

The conventional CPP algorithm has been applied to a fully-connected road-network for use by ground vehicles. However, since UAVs do not have any restrictions such that they must only move along the roads, the CPP algorithm needs to be modified for the case that UAVs search a general roadmap having unconnected road segments. The modified CPP (mCPP) generates a tour of the road-network traveling all the roads once no matter what the type area of interest map is: an even or odd graph. Even searching the area having no road somewhere in it can be tackled by the mCPP algorithm by generating a virtual road pattern with a lawnmower [75] or spiral-like [76] algorithm. Figure 3.2 exemplifies a sample road-network search problem to be solved by the mCPP.

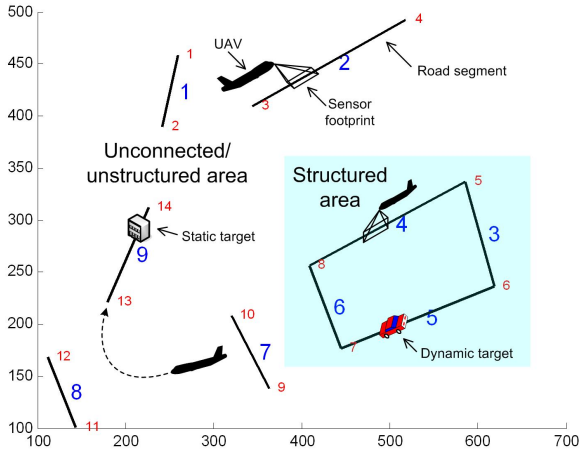


Figure 3.2: An illustration of the modified Chinese Postman Problem (mCPP)

As the CPP algorithms generally approximate their path to a straight line for simplicity, the mCPP algorithm using a straight line path is called the modified Euclidean CPP (mECPP) for the rest of this chapter. However, in order to produce the shortest path flyable by the UAV that connects the road segment sequence selected by the search route algorithm, the flight constraints of the UAV have to be taken into account. To accommodate this, the Dubins path is incorporated into the mCPP algorithm instead of using just a straight line to connect the roads, and this is called the modified Dubins CPP (mDCPP). It is worthwhile noting that the mDCPP accommodates an approach angle constraints to a road segment by using the Dubins paths unlike the Dubins traveling salesman problem (DTSP) [69].

3.3 Road-Network Search Route by Multiple Unmanned Aerial Vehicles

Since the UAV cannot change its heading angle instantaneously due to the physical constraint as shown in Fig. 3.3, its minimum turning radius should be taken into account explicitly to design a road-network search route pattern. Moreover, when visiting a certain edge unconnected to the main road-network or when coming back to the base, UAVs do not have to fly over the road as explained in the previous section. Therefore, this section proposes road-network search route

planning algorithms for the mCPP using multiple UAVs. In the first place, the optimisation via MILP is introduced and incorporated with the Dubins path. A new approximation algorithm is then proposed as a more practical approach to reduce the complexity of the algorithm.

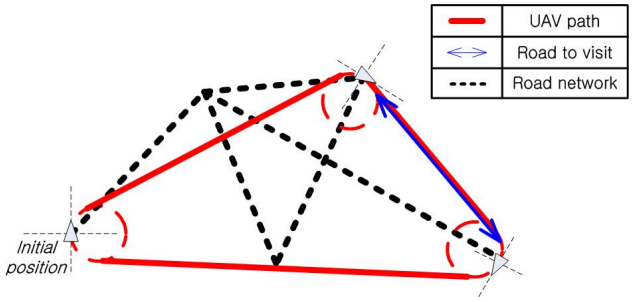


Figure 3.3: An illustration of the road-network search by a UAV

3.3.1 Optimisation via MILP

The mCPP is first solved by the MILP optimisation using the MMKP formulation to find an sub-optimal solution minimizing the total flight time of UAVs. A classical MMKP is to pick up the items for the knapsacks having a maximum total value so that the total resource required does not exceed the constraints of knapsacks [77]. For applying the MMKP to the road-network search, UAVs are assumed as the knapsacks, the roads to be searched as resources, and the limited flight time or energy of each UAV as the capacity of knapsack. The MMKP formulation allows the search problem to consider the characteristics of each UAV such as flight time capacity and minimum turning radius. Moreover, since the Dubins path planning produces the shortest and flyable paths taking into consideration their dynamical constraints, it is used to calculate the cost function of the MMKP. The details of the proposed road-network search algorithm for multiple UAVs are explained as follows.

3.3.1.1 Generation of the shortest edge permutation

First of all, unordered feasible edge (that is, road) permutations to be visited by the UAV are generated for all possible cases with a given petal size. The petal

size means the maximum number of edges that can be visited by one UAV and is determined by the amount of resources available to it. As a general description, let us assume that the edge set E is $\{e_1, e_2, e_3\}$, and the petal size is three, then all possible edge permutations can be generated as:

$$E_p = [e_1,\ e_2,\ e_3,\ \{(e_1, e_2),\ (e_2, e_1)\},\ \{(e_1, e_3),\ (e_3, e_1)\},\ \{(e_2, e_3),\ (e_3, e_2)\},\ \{(e_1, e_2, e_3),\ (e_1, e_3, e_2),\ (e_2, e_1, e_3),\ (e_2, e_3, e_1),\ (e_3, e_1, e_2),\ (e_3, e_2, e_1)\}]$$

If the end vertex of one edge $e_1 \in E_p$ and any vertex of next edge $e_2 \in E_p$ are not connected, they are connected with an additional edge which has a shorter distance. Then, the shortest order-of-visit edge permutations considering the initial position of each UAV are computed under the assumption that a path is represented as a straight line. The example of the shortest order-of-visit permutation $E_{p,s}$ among the aforementioned edge permutation set E_p are as:

$$E_{p,s} = \{e_1,\ e_2,\ e_3,\ (e_2, e_1),\ (e_1, e_3),\ (e_2, e_3),\ (e_2, e_3, e_1)\}$$

3.3.1.2 Dubins path planning

Once the shortest edge permutations are determined, the next step is to compute and to store the cost (path length or flight time) of them. In this step, this chapter uses the Dubins path which is able to take into account the orientation and path constraints of the UAV instead of just using Euclidean distance of each edge. The Dubins path is the shortest path connecting two configuration represented by position and pose under the constraints of a bound on turning radius. The Dubins path is formed either by concatenation of two circular arcs with their common tangent (CSC path) or by three consecutive tangential circular arcs (CCC path) as derived in Section 2.1. Although the CSC path is being used in general case, this study explores the CCC path as well as the CSC path for the search route planning within a densely distributed road environment because CSC path cannot be applied to a pair of positions having a distance less than the minimum turning radius. Moreover, to follow the road precisely taking into account the sensor footprint coverage, the path should consist of both CCC and CSC forms of Dubins path. Figure 3.4 shows an example of a road search path using CSC

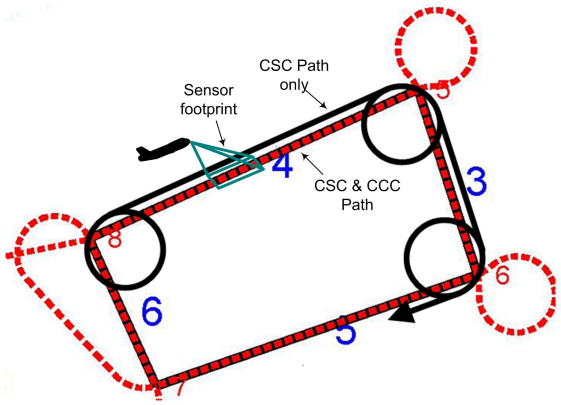


Figure 3.4: An example of CCC and CSC paths following the road

and CCC path for a small sensor footprint, which results in a detour at the road intersection. Note that using only CSC path cannot generate the path exactly following the road as black solid line due to turning constraints.

3.3.1.3 MMKP formulation and MILP optimisation

The final step of the proposed algorithm is allocating the edge permutations to each UAV so as to cover all the roads with a minimum flying time. This can be expressed by a MMKP formulation as:

$$\min\ J = \sum_{i=1}^{N_{UAV}} \sum_{j=1}^{N_{p_i}} T_{ij} x_{ij} \tag{3.1}$$

$$\text{s.t} \sum_{i=1}^{N_{UAV}} \sum_{j=1}^{N_{p_i}} E_{kj} x_{ij} \geq 1,\ \text{for } k = 1, \ldots, N_{edge} \tag{3.2}$$

$$\sum_{j=1}^{N_{p_i}} x_{ij} = 1,\ x_{ij} \in \{0, 1\}, \tag{3.3}$$
$$\text{for } i = 1, \ldots, N_{UAV},\ j = 1, \ldots, N_{p_i}$$

where N_{UAV}, N_{edge}, and N_{p_i} denote the number of UAVs, edges to be visited and permutations generated by the i-th UAV, respectively. T_{ij} represents mission cost (flight time) of j-th permutation of i-th UAV and E_{kj} represents the matrix whose k-th element of j-th permutation is 1 if edge k visited, otherwise 0 and x_{ij} is either 0, implying permutation j of the i-th vehicle is not picked, or 1 implying permutation j of the i-th UAV is picked. First constraint represents that UAVs

should visit every edge once or more and second one represents the allocation of the exact one edge permutation to the each UAV. This MMKP problem is solved by SYMPHONY MILP solver [78]. It should be noted that depending on the petal size, the computational burden of the mission cost T_j of all edge permutations would be increased significantly. Moreover, obtaining a solution to any MILP formulation is NP (Non-deterministic Polynomial-time) hard, and it is claimed that problems in which the sum of the number of vehicles and targets is less than 12 are solvable in less than a minute on a modern desktop computer [39, 79]. Therefore, large problems with complex road-network will require a reformulation of the problem such as restricting the petal size and using receding horizon concept [80] or partitioning of the road-network in a timely manner to use the MILP optimisation approach.

3.3.2 Approximation Algorithm

3.3.2.1 Nearest insertion based mDCPP

Due to the complexity of the problem, instead of using the optimisation method explained above, an approximation algorithm is developed as a more practical way for the mCPP. To develop the approximation algorithm, the TSP algorithm was first studied. Although there are a lot of algorithms for the TSP [81], one heuristic approach, a nearest insertion method is adopted since it is fast and easy to implement. The basic idea of it is to construct the approximation tour by a sequence of steps in which tours are constructed for progressively larger subsets of the nodes. It produces a tour no longer than twice the optimal regardless of the number of nodes in the problem and runs in a time proportional to the square of the nodes [81]. Having these in mind, this section proposes the nearest insertion based mDCPP (NI-mDCPP) algorithm. Firstly, the NI-mDCPP algorithm for the single UAV is described as follows.

Algorithm description

1. Start from a certain point or road junction and select the nearest road to it using the Dubins path length.

2. Make and grow a tour by finding the nearest road to any of the selected tour roads.

3. Compute the cost of insertion to decide whether to insert before or stack after the closest road to the tour.

4. Insert the selected road in the decided position.

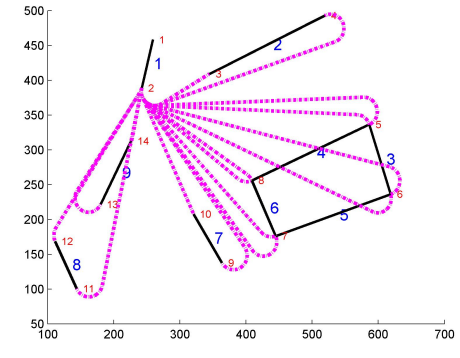
5. 2 ~ 4 are repeated until all roads are included in the tour.

3.3.2.2 Euclidean distance order approximation

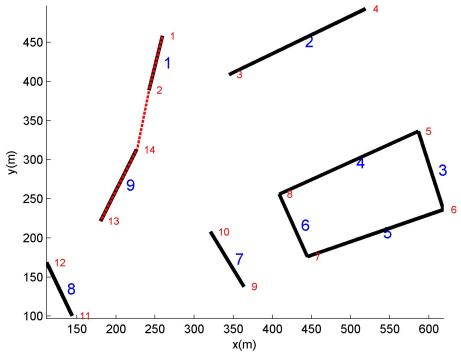
To reduce the computation burden further for the dynamic environment, an additional approximation algorithm which uses the Euclidean distance order is incorporated into the NI-mDCPP algorithm. This algorithm is described as follows. First of all, make the ascending order of road list for both the Euclidean distance and the Dubins path length between all pairs of end points of the road-network, and find the maximum number, $n_{order,max}$ which guarantees that road list of Euclidean distance within that number contains the shortest Dubins path in advance. Note that although $n_{order,max}$ is determined before running the algorithm with given information of the road-network, a size (tendency) of $n_{order,max}$ would remain almost the same against minor changes of road information for an uncertain dynamic environment. Then, when finding the nearest roads, make the ascending order list of distance between edge of interest and all the other roads with Euclidean distance first, and find the nearest road whose Dubins path length is the shortest among roads in the $n_{order,max}$ Euclidean distance order. In other words, this method computes only $n_{order,max}$ Dubins path distances instead of computing all Dubins lengths between one road to the others. For example, if $n_{order,max}$ is three as shown in table 3.1, among ascending road list of Euclidean distance from a certain road to the others, only first three roads (3, 6 and 4) are tested by the Dubins path, and the shortest one (6) is selected as the nearest road. Note that as the distance between the roads increases, since the Euclidean distance and Dubins length gets closer, $n_{order,max}$ decreases resulting in less computation time.

3.3.2.3 Negotiation for multiple UAVs

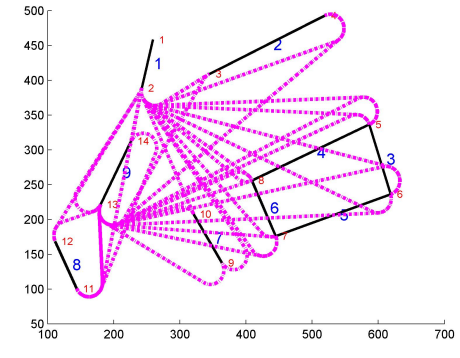
Having the characteristic of the NI-mDCPP for the single UAV, the algorithm can be extended to the case of multiple UAVs using auction-based negotiation. The



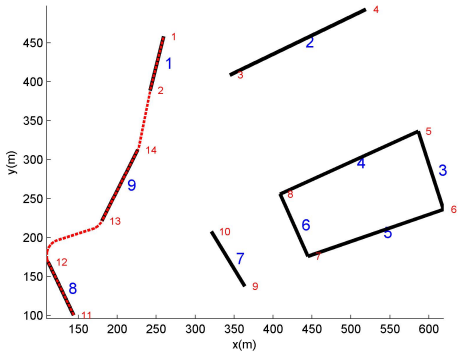
(a) Find the nearest edge



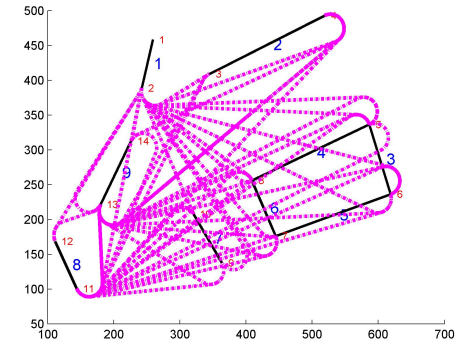
(b) First insertion



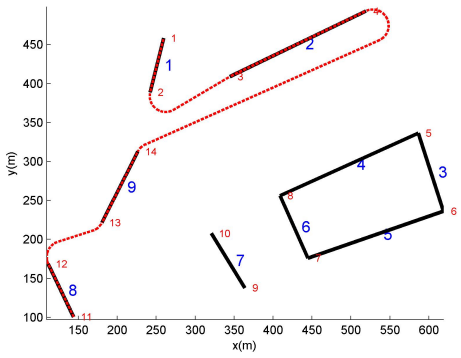
(c) Find the nearest edge



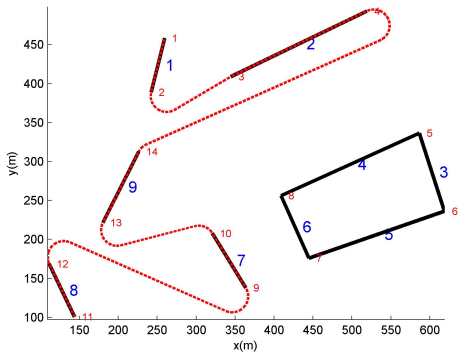
(d) Second insertion



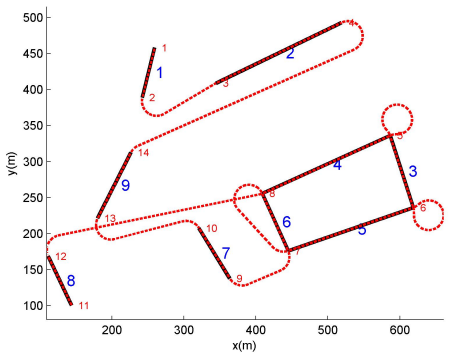
(e) Find the nearest edge



(f) Third insertion



(g) Fourth insertion



(h) Eighth insertion

Figure 3.5: An example simulation of the NI-mDCPP road search algorithm

Table 3.1: Procedure of Euclidean distance order approximation algorithm ($n_{order,max} = 3$)

Euclidean				Dubins		Nearest Road
Order	Length	Road		Length	Order	
1	1.2	**3**	→	3.1	3	
2	1.3	**6**	→	2.8	1	**6**
3	1.4	**4**	→	2.9	2	
4	1.5	2				
5	1.6	5				
6	1.7	7				
⋮	⋮	⋮				

auction algorithm is a method of solving the assignment problem using elements from both the primal and dual formulations based on approximate optimality, called *ϵ-complementary slackness (CS)* condition [82]. Suppose that there is a benefit a_{ij} for matching an agent i and an object j, object j has a price p_j, and the person who receives the object must pay the price. Let us then set a positive scalar ϵ and state that an agent i is almost happy with an assignment and a set of prices if the value of its assigned object j_i is within ϵ of being maximal as:

$$a_{ij_i} - p_{j_i} \geq \max_{j=1,\ldots,n} \{a_{ij} - p_j\} - \epsilon \tag{3.4}$$

Above ϵ-CS condition is interpreted as stating if an agent i is assigned to object j, then its profit is equal to the benefit it receives from that object minus its price, and that object j provides an agent i more profit (within some tolerance ϵ) than any other object. The auction algorithm starts with an empty assignment (which trivially satisfies ϵ-CS), and iteratively proceed by modifying it through bidding (decision is based on values, which are bidder's benefits plus bidding increment ϵ at each time) and assignment phase until all agents are assigned satisfying ϵ-CS. This auction process is proved to be terminated in finite number of rounds, and the performance in terms of computation and optimality depends strongly on the size of ϵ [82].

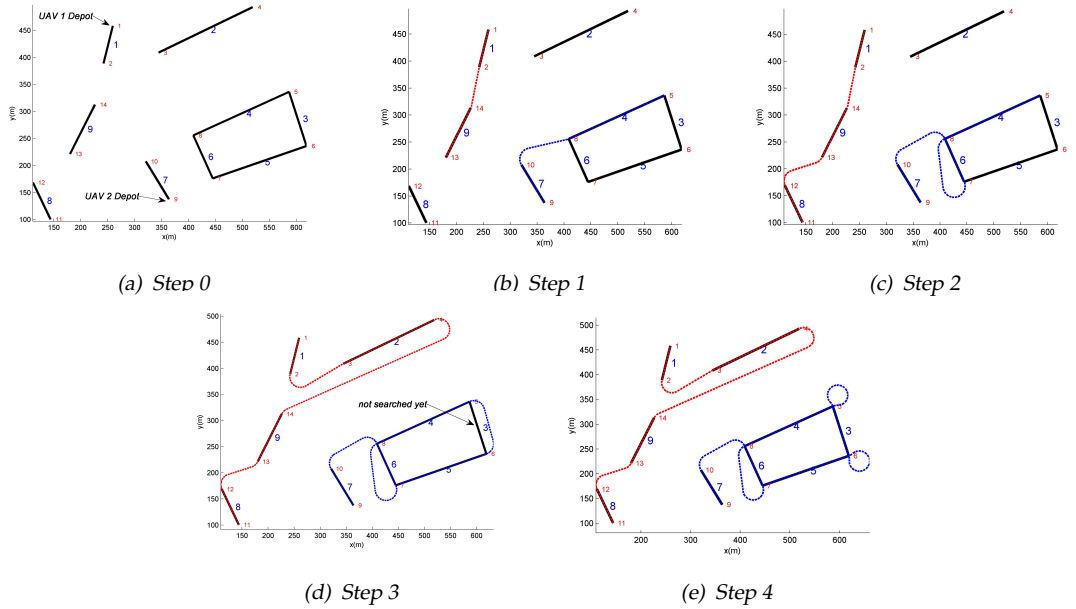
In this study, the auction benefit a_{ij} of UAV i for road segment j is its corresponding flight time to that road segment. The auction algorithm is used to optimally assign roads to UAVs since it is known to be intuitive and efficient com-

pared to the other assignment algorithms [83]. The auction negotiation process is performed in a decentralised way by sending and receiving bidding information amongst each other while operating as both auctioneer and bidder accordingly. To avoid the bidding increment being zero when several UAVs compete for the same road thereby creating an endless cycle, above type of auction algorithm is used. In addition, this competitive auction would be beneficial when there is uncertainty in flight time (cost) or noise during communication as UAVs competitively bid for the best solution with currently available information. Now, the NI-mDCPP algorithm for multiple UAVs is described as follows.

Algorithm description

1. Start with N initial positions or roads of N UAVs.
2. Make and grow a tour by using single NI-mDCPP algorithm while storing the cost (path length or flight time) between selected tour roads and remaining edges (which was needed for finding the nearest vertex).
3. When conflict occurs, i.e. more than one UAV wants the same road for the next tour, the auction algorithm using stored cost is used to match UAVs with the task (road) to minimise the cost.

Figure 3.6 illustrates the procedure of the algorithm. Since road 3 is not searched yet in Fig. 3.6(d), each UAV sends its cost for the given task (in this case, visiting road 3), then, auction or bipartite (linear programming) is used to match UAVs with the task to minimise the cost. Although overlapping road segments is avoided using the auction algorithm, collision between UAVs might occur during the transition from one to another road. In other words, the distance between any two UAVs can be less than a threshold value, called as minimum separation distance. In that case, if necessary, the path can be replanned either by visiting new road or varying the curvature of the arc of the Dubins path. For simplicity, this study assumes that the collision avoidance is done by local flight controller or operating UAVs in different altitudes. The proposed algorithm is rather simple but straightforward and can be run in real time. Moreover, by including additional factors such as different minimum turning radius and total path length (or the number of roads) assigned to each UAV so far into the cost, the auction-based negotiation can be varied flexibly, even for heterogeneous UAVs.



(a) Step 0 (b) Step 1 (c) Step 2

(d) Step 3 (e) Step 4

Figure 3.6: Negotiation procedure for multiple UAVs

3.4 Numerical Simulations

3.4.1 Monte Carlo Simulation of the NI-mDCPP

To evaluate the properties and performance of the proposed approximation algorithm, Monte Carlo simulations are performed using a random map with different parameters about the map size and the number of UAVs. A map environment is composed of 10 by 10 vertices as numbered in Fig. 3.7, and road edges are generated by connecting two vertex randomly selected. To check the impact of the map size on Dubins path planning, the distance d_{map} between the adjacent vertex is set to be in proportion to the minimum turning radius ρ_{min} of the UAV as:

$$d_{map} = K_s \times \rho_{min} \tag{3.5}$$

where K_s is the scale factor. Moreover, some of the selected edges whose lengths are longer than three times of d_{map} are discarded to get a well-distributed road-network and to distribute the roads to each UAV with a similar length. The threshold for the discard can be varied depending on the map environment. Figure 3.7 shows the sample road-network with 20 randomly chosen edges. By

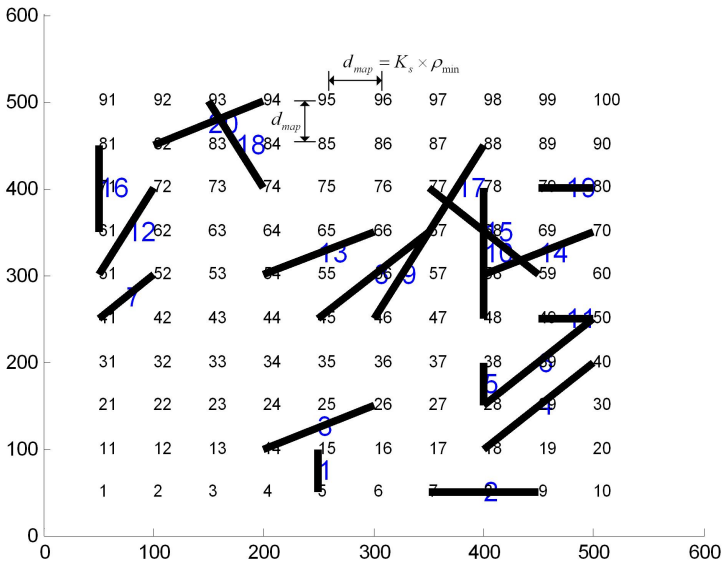


Figure 3.7: A sample road-network with 20 randomly chosen edges ($K_s = 1$, $\rho_{min} = 50\ m$)

Monte Carlo simulations, the effect of three major factors for the road-network search route planning can be investigated. These are:

The distribution density of road-network: Densely or sparsely distributed road relative to the minimum turning radius of UAVs

The type of path planning: Straight line or Dubins path

The number of UAVs: Computation time, path length, and the longest path length of a single UAV

This will provide information on priorities for the efficient use of the UAV group in the planning phase of the autonomous search mission. In the simulations, UAVs are assumed to have a constant velocity and minimum turning radius $\rho_{min} = 50\ m$. The simulation results are the average of 50 runs with the MATLAB algorithm. The same sequence of randomly generated environments is used for each algorithm.

3.4.1.1 Single UAV case

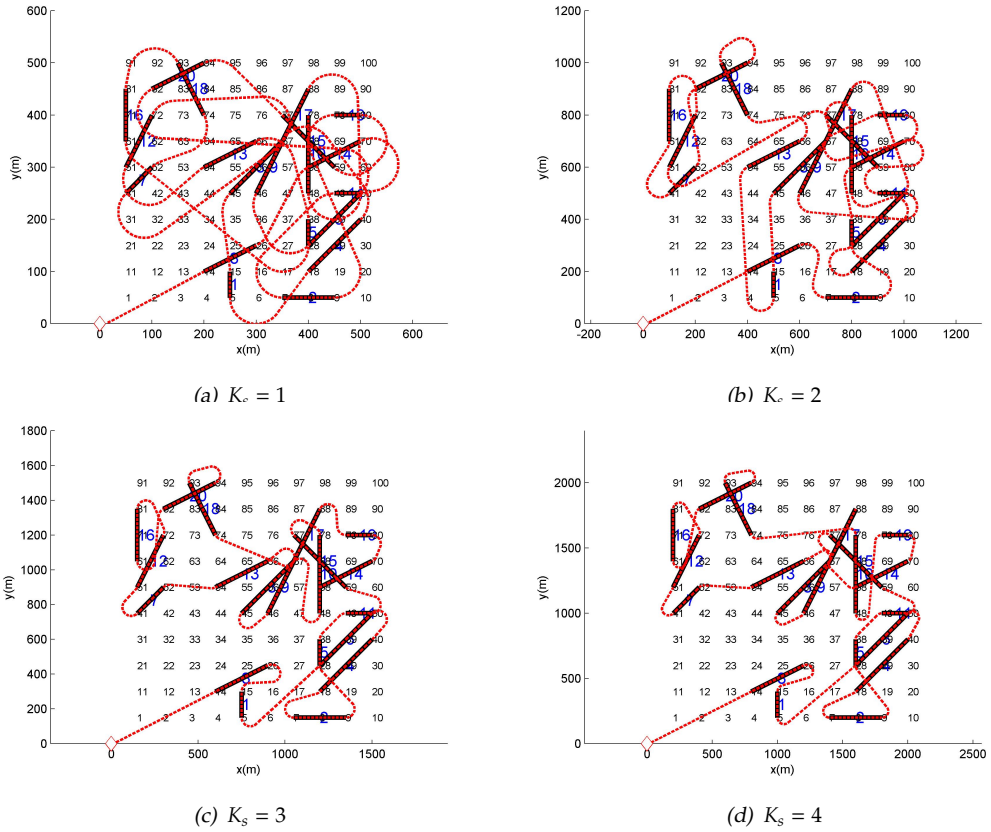
The first set of the simulation is performed by using a single UAV with different road map scales. For the rest of this section, the terms of mDCPP and the mECPP

are used for the NI-mDCPP and the NI-mECPP, respectively. One of search route planning results using the mDCPP for the random map is shown in Fig. 3.8, which covers all the roads satisfying turning constraints of UAVs. Figure 3.9(a) displays the computation time ratio between the Dubins path (mDCPP) and the straight line (mECPP). Regardless of the map scale, the mDCPP algorithm is around 35 times slower than the mECPP constantly. Meanwhile, the computation time of the mDCPP along with the Euclidean distance order approximation decreases as the map scale increases resulting from decrease of the maximum order $n_{order,max}$ as shown in Fig. 3.9(b).

Figure 3.9(c) compares total path length to cover the entire roadmap using the mDCPP and the mECPP. For fair comparison, the length of the mECPP (denoted by L^*_{mECPP}) is computed by road search route from the mECPP algorithm but connecting roads using Dubins path. This is because although road search route planning is performed by the mECPP, a real trajectory of the UAV connecting roads should be the Dubins type of path restricted by its maximum curvature. When the minimum turning radius is relatively small compared to the distance between roads, that is, when the map scale is small, the path length of the mDCPP is shorter than that of the mECPP. However, as the map scale gets bigger, the path length ratio gets closer to nearly one (or even over one) since the road search order of the Dubins path would be almost the same as the one from the straight line as one can expect from Fig. 3.8. Based on these analysis on the computation time and the path length advantage, it is concluded that using the Dubins path algorithm is preferable for the road-network searching by a single UAV in case that the map scale is smaller than two.

3.4.1.2 Multiple UAVs case

In the previous section, the performance of the proposed NI-mDCPP algorithm was compared to that of the NI-mECPP as well as NI-mDCPP without Euclidean approximation for single UAV. Now, let us compare the performance of algorithms for multiple UAVs case with different road map scales. The results are shown as the normalised value to present the effect of changing the number of UAVs fairly between factors investigated. One example of the search route planning results using the mDCPP for six UAVs is represented in Fig. 3.11. The initial position of each UAV is equally distributed around the road area. Figure 3.10 presents



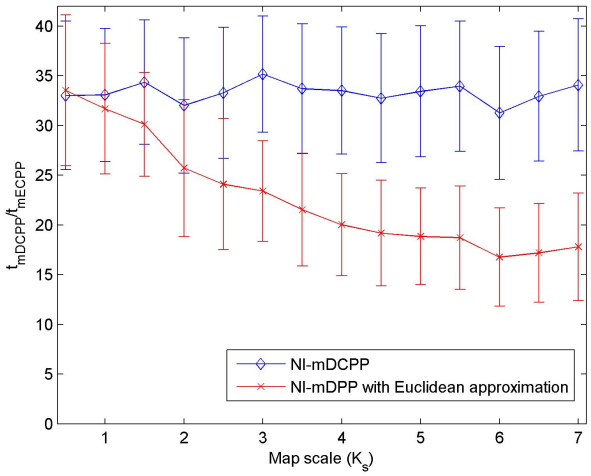
(a) $K_s = 1$ (b) $K_s = 2$

(c) $K_s = 3$ (d) $K_s = 4$

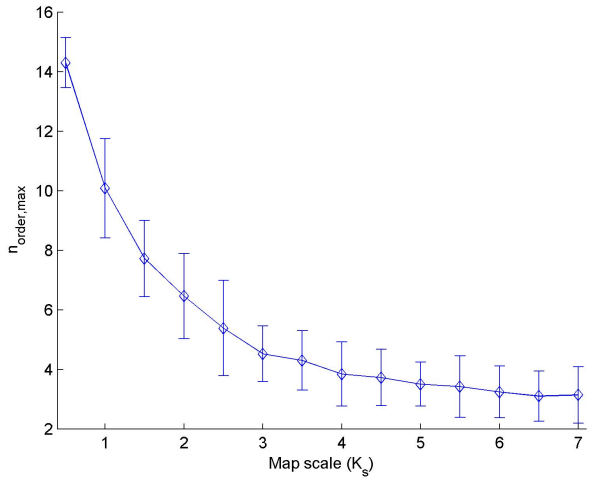
Figure 3.8: NI-mDCPP road search path with different map scale factors

the normalised simulation results by the value of a single UAV. In particular, the longest path length (Fig. 3.10(c)) of the UAV is investigated since it is equivalent to the mission completion time of the entire UAV team. The normalised computation time and the longest path length of the UAV are decreased as the size of the UAV team increases regardless of the map scale since each UAV takes partial charge of the road search mission cooperatively using the auction-based task assignment. Unlike the others, the total path length (Fig. 3.10(b)) is affected by the map scale significantly. When the map scale is small, the total path length is decreased in proportional to the number of UAVs. Whereas in a relatively big map environment, the normalized path length remains nearly at one since each UAV should fly a long distance from the initial position or one road to the another road.

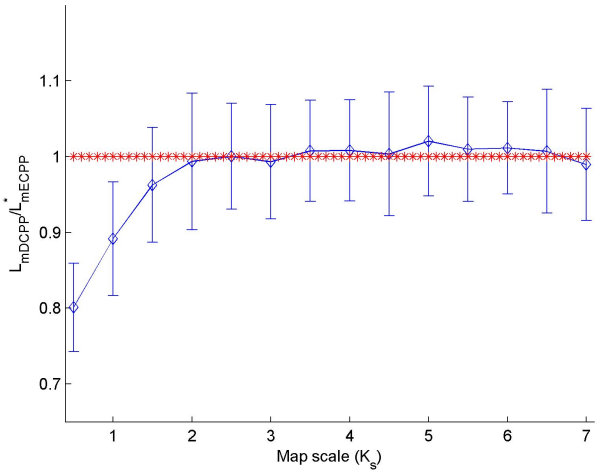
Apparently, the simulation results show that the bigger the UAV team size is, the better performance it shows in term of the computation time and path length. However, using a large size of the UAV team requires more operational



(a) Computation time ratio



(b) $n_{order,max}$ for Euclidean approximation



(c) Path length ratio

Figure 3.9: NI-mDCPP results with different map scale factors, average for 50 simulations

cost. Therefore, the performance index to determine the optimal size of the UAV team for the search mission can be proposed as Eq. (3.6) including an additional operational cost on each UAV represented as the normalised number of UAVs (by maximum seven UAVs in this chapter), $\bar{n}_v$, and its weighting factor, w_{n_v}.

$$J = w_t\bar{t} + w_l\bar{l} + w_L\bar{L} + w_{n_v}\bar{n}_v \tag{3.6}$$

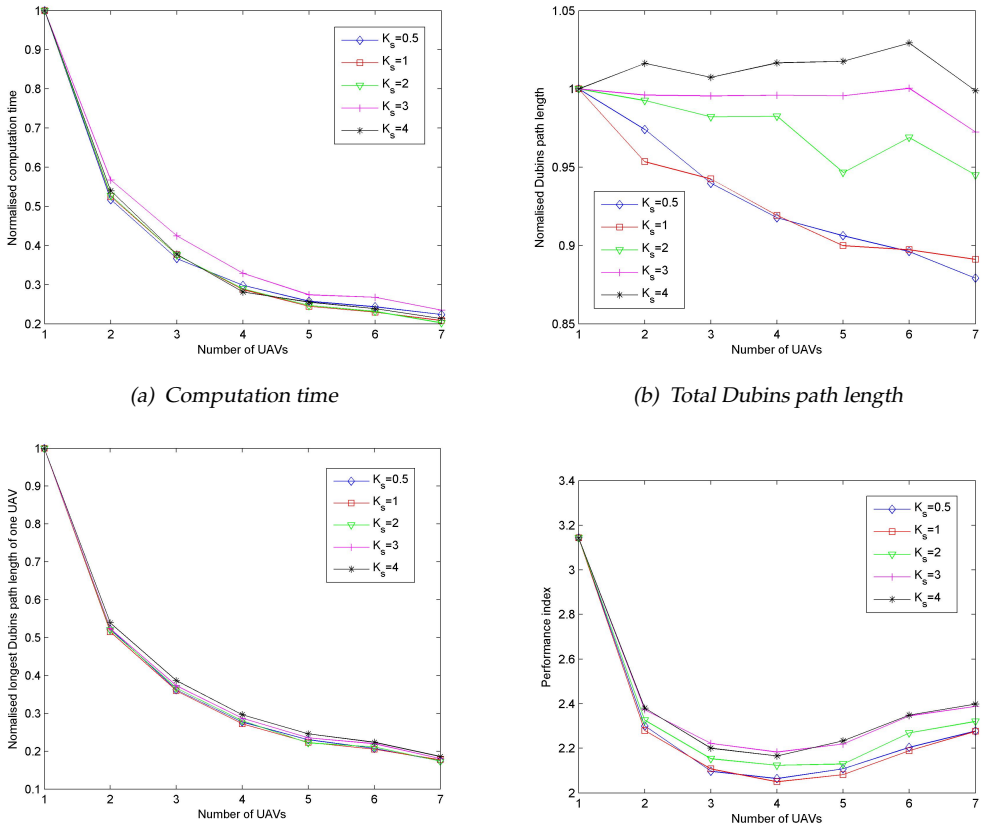
where w_t, w_l and w_L represent the weighting factors of computation time, total path length and the longest path length of one UAV, respectively. Under the assumption that all the weighting factors are equally one, the number of UAVs to minimise the performance index J can be selected as four for all map scale factors ($K_s = 0.5 \sim 4$) consistently as shown in Fig. 3.10(d). Even though this sub-optimal number of UAVs can be changed according to different map or operational parameters, since the result comes from Monte carlo simulations with random maps with various map scale factors, this framework would help the operator to decide the reasonable number of resources (UAVs) in the initial phase of autonomous search mission. Besides, since this study used a simple operational cost in proportion to the number of UAVs, more rigorous approach to capture and incorporate the realistic operational cost should be followed depending on the characteristic of the mission and the vehicle such as fuel, communication resources, or surrounding environments as future work.

3.4.2 Performance Comparison

To evaluate the performance of the proposed road-network search algorithms for multiple UAVs, numerical simulations are performed for a specific scenario with four UAVs and the road-network given in section 3.2. Each UAV has different dynamic constraints as:

- Minimum turning radius ρ_{min}: [100 90 80 70] m
- Maximum cruise speed $V_{c,max}$: [60 50 40 30] m/s

The maximum curvature κ_{max} of UAVs can be given by $\kappa_{max} = 1/\rho_{min}$. UAVs are assumed to have maximum cruising speed during the entire mission and the maximum petal size of the edge permutation is set to be five. Figure 3.12(a) shows



(a) Computation time

(b) Total Dubins path length

(c) The longest Dubins path of one UAV

(d) Performance index cost

Figure 3.10: NI-mDCPP results with different number of UAVs, average for 50 simulations

the result of the road-network search using the MILP optimisation. The flight path is smooth and flyable due to the Dubins path planning, and since the UAV does not need to fly along the road only, the results include additional paths connecting some of the unconnected roads. Moreover, it can be observed that the faster UAV tends to have a longer flight path since the objective of the MILP is to minimise the total sum of flight time of all UAVs. The total flight length of all UAVs is 2798.1 metres, and its flight duration is 583.1 seconds. In this scenario, the total computation time exceeds a reasonable limit (> 5 minutes) using Matlab for the MMKP formulation and ANSI-C for the MILP computation using a normal PC system (Core 2 CPU, 2.0 Ghz, and 512 MB RAM). Meanwhile, Fig. 3.12(b) shows the search result of the NI-mDCPP using the cost of the flight time.

Table 3.2 shows the performance comparison between MILP, NI-mECPP, NI-mDCPP* (without Euclidean approximation) and NI-mDCPP algorithms. The maximum number $n_{order,max}$ is set to six for this sample map environment. Note

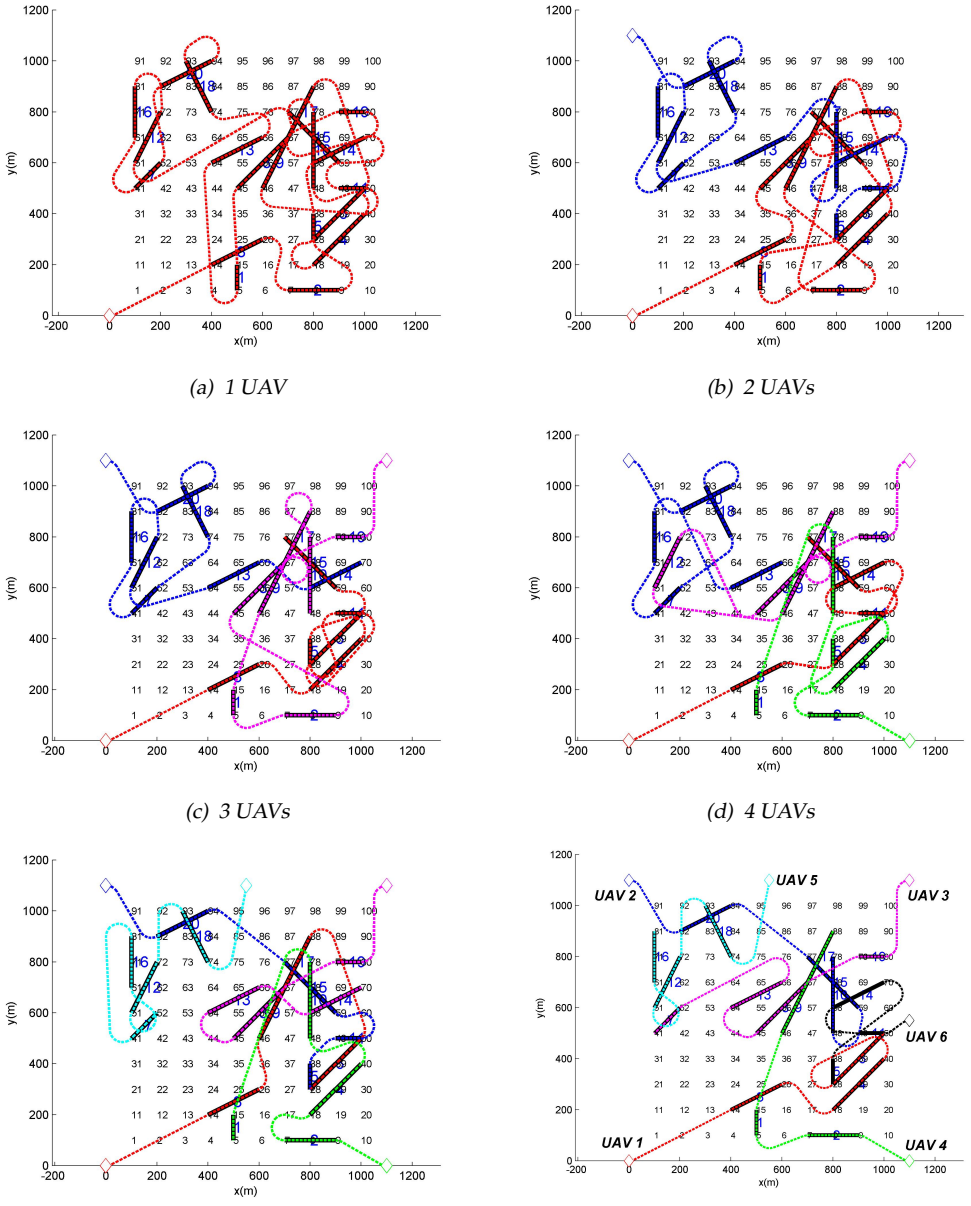
Table 3.2: Performance comparison between different algorithms

Method	Computation time	Flight time (sec)
MILP	> 5 min	583.1
NI-mECPP	0.33 sec	681.0
NI-mDCPP*	1.15 sec	642.9
NI-mDCPP	0.78 sec	642.9

that the flight time of the NI-mECPP is computed by road search route from the NI-mECPP algorithm but connecting roads using Dubins paths with corresponding minimum turning radius. The proposed NI-mDCPP gives a solution within a second having about ten percent longer flight time (642.9 seconds) than that of the MILP optimisation. Considering both computation time and performance, the proposed NI-mDCPP can be regarded as a preferable approach over the MILP optimisation for a given sample map or more complex scenarios in case that a rapid decision needs to be made. Note that this preference or optimality of performance related with combination of computation and flight time varies depending on the mission requirements.

3.5 Summary

This chapter has presented practical approaches for a road-network search strategy by which a team of UAVs visit every road in the map of interest. The conventional CPP algorithm was explained and modified for the general type of roadmap including unconnected roads. Then, the MILP optimization and the nearest insertion algorithm along with the auction negotiation were proposed for multiple UAVs. To realistically accommodate the manoeuvreing constraints of UAVs, the Dubins path planning was used for solving the modified CPP. The performance of the proposed algorithm was investigated via a Monte Carlo simulation by analyzing the effects of different map sizes, path planning methods and the number of UAVs. Moreover, to clarify the benefit of the proposed algorithm, this chapter compared the performance of the optimisation and approximation algorithm in terms of computational load and flight time for a specific real road-network.

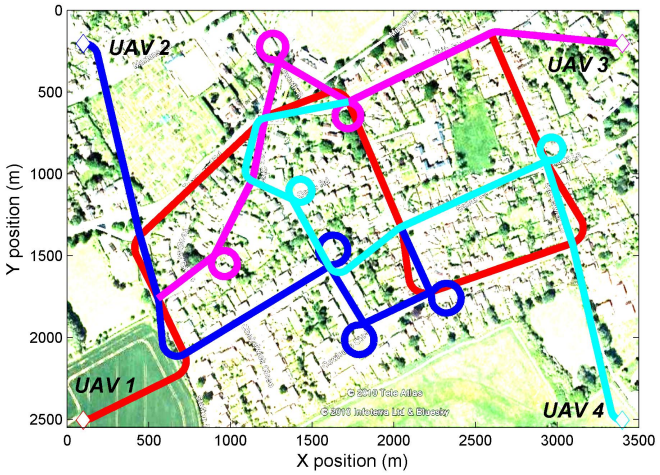


(a) 1 UAV (b) 2 UAVs

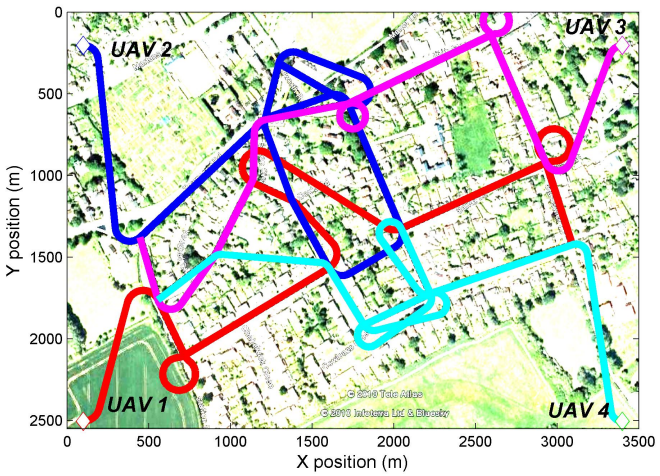
(c) 3 UAVs (d) 4 UAVs

(e) 5 UAVs (f) 6 UAVs

Figure 3.11: NI-mDCPP road search path with different number of UAVs ($K_s = 2$)



(a) MILP optimisation



(b) NI-mDCPP

Figure 3.12: Road-network search route planning results using multiple UAVs

Chapter 4

Ground Moving Target Tracking With Information Fusion

This chapter presents ground moving target estimation algorithms for the UAV surveillance problem as a background of this thesis. To produce appropriate surveillance data to be used by UAVs, a GMTI (Ground Moving Target Indicator) is a well-suited radar sensor due to its wide-coverage, all-weather, day/night, and real-time capabilities [84]. A VMTI (Visual Moving Target Indicator) sensor in both visible and infrared bands has also been developed recently to automatically detect and track moving targets on the ground [2]. From this sensor data such as range, azimuth, or elevation of the target with respect to the sensor location (along with appropriate target dynamics), a certain level of accurate estimation could be obtained using conventional filtering techniques. However, as ground target tracking is a challenging problem due to the uncertainty of the target manoeuvre, all available information sources should be exploited: its own sensor data, data from other UAVs, and contextual knowledge about the sensor performance and the environment. In other words, information fusion is required to improve the estimation accuracy.

In particular, in many applications for ground target tracking, the majority of ground vehicles are moving on road networks whose topographical coordinates could be known with a certain accuracy. Such road-map information can be used for improving the quality of tracking significantly by constraining the state of the ground target interested especially in its position, velocity and acceleration within the road geometry. This is known as road-constrained target tracking problem, and largely, there are two categories of techniques for making use of the information about the roads. The first one is the post-processing correction techniques, which runs tracking algorithm first without the road information, and correction is then applied. Tang *et al.* [33] and Kanchanavally *et al.* [85] proposed bayesian filtering method with the hospitability map which is a map providing a likelihood for each point proportional to the ability of a target to move at that location. Along with this approach, Kassas *et al.* [86] added the concept of synthetic inclination

map which describes how the target will be synthetically inclined to move in different directions with certain velocity component. The second one is based on the Kalman filtering framework. Dan *et al.* [87] proposed Kalman filtering with state equality constraints and used road information as equality constraints. Zhang *et al.* [88] used pseudo measurement approach which treats the road constraints as additional fictitious measurement based on the work of Tahk and Speyer [89]. Herrero *et al.* [34] proposed the preprocessing of sensor measurements with map restriction. Moreover, they introduced map-tuned interactive multiple model (IMM) structure which consists of constant speed, longitudinal acceleration model, and curvilinear model incorporating map information. To deal with road network which has road junctions and crossing of several roads, variable structure IMM filtering concept is also proposed by [32, 90]. Even though the particle filter might result in better tracking performance depending on the situation, particularly for highly nonlinear system and non-Gaussian noise as described in [91, 92], it would require a significant computational cost. Since we consider the use of small and low-cost UAVs rather than a single UAV with high computation power, this study mainly utilises Kalman filter-based algorithms.

This chapter first presents the conventional target tracking using an acceleration model as a realistic target dynamics and the simplified measurement model of the VMTI and GMTI. Then, a practical road approximation algorithm is introduced using constant curvature segments. To exploit this road information for precise target estimation, nonlinear road-constrained Kalman filtering is applied using a pseudo-measurement approach. For behaviour classification of a ground vehicle presented in the Chapter 5, an optimal smoothing technique is also presented, which uses the past state estimates to enhance the tracking accuracy. Lastly, sensor fusion techniques are explained considering that multiple UAVs equipped with an onboard sensor are involved to track the same target simultaneously.

4.1 Ground Target Model

General target tracking filters have traditionally been developed for monitoring aerial targets such as airplanes, missiles and so on [93]. Although ground vehicles move with much lower speeds than aerial targets, they often perform irregular

stop-and-go manoeuvres with a much smaller turn radius. A constant velocity model usually used for radar target tracking is thus unsuitable for tracking ground vehicles, and hence an acceleration or jerk model is a more suitable model. After analyzing the car trajectory data acquired by running an S-Paramics traffic simulation program [94] (Fig. 4.1) and considering general driving behaviour, it is observed that the jerk is not negligible, with the acceleration best modelled using a piecewise constant profile over a specific duration of time, as shown in Fig. 4.2.

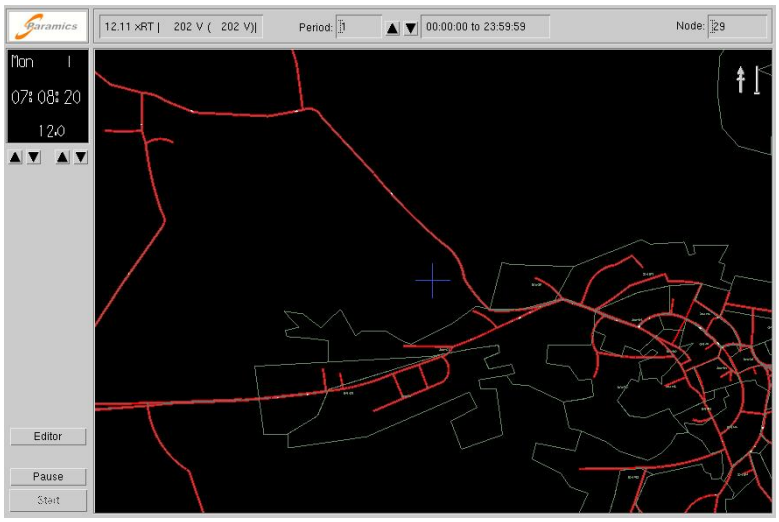


Figure 4.1: S-Paramics model of Devizes, Wiltshire, UK

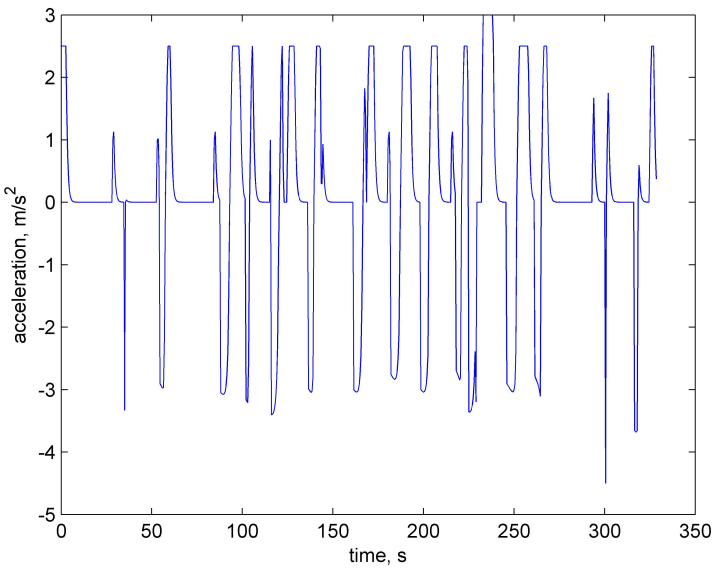


Figure 4.2: Acceleration data of a ground vehicle in S-Paramics Devizes model

Hence, a good model to apply to the tracking of ground targets considers acceleration dynamics [95]. This acceleration model defines the target acceleration as a correlated process with a decaying exponential autocorrelation function, which means if there is a certain acceleration rate at a time t then it is likely to be correlated via the exponential at a time instant t+τ. A discretised system equation for the acceleration model for a ground vehicle is thus expressed in the form:

$$\mathbf{x}_{k+1}^t = F_k \mathbf{x}_k^t + \eta_k \tag{4.1}$$

here the state vector is $\mathbf{x}_k^t = (x_k^t, \dot{x}_k^t, \ddot{x}_k^t, y_k^t, \dot{y}_k^t, \ddot{y}_k^t)^T$, and where η_k is a process noise which represents the acceleration characteristics of the target. The state transition matrix F_k is given by [95, 96]:

$$F_k = \begin{bmatrix} 1 & T_s & (e^{-\alpha T_s} + \alpha T_s - 1)/\alpha^2 & 0 & 0 & 0 \\ 0 & 1 & (1 - e^{-\alpha T_s})/\alpha & 0 & 0 & 0 \\ 0 & 0 & e^{-\alpha T_s} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & (e^{-\alpha T_s} + \alpha T_s - 1)/\alpha^2 \\ 0 & 0 & 0 & 0 & 1 & (1 - e^{-\alpha T_s})/\alpha \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha T_s} \end{bmatrix} \tag{4.2}$$

where α is a correlation parameter which models different classes of targets: a small α for targets with relatively slow manoeuvres and a high α for targets with fast and evasive manoeuvres. The covariance matrix of the process noise η_k can be modelled as follows:

$$Q_k = V[\eta_k] = \frac{\sigma_a^2}{\alpha^4} \begin{bmatrix} q_{11} & q_{12} & q_{13} & 0 & 0 & 0 \\ q_{12} & q_{22} & q_{23} & 0 & 0 & 0 \\ q_{13} & q_{23} & q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{11} & q_{12} & q_{13} \\ 0 & 0 & 0 & q_{12} & q_{22} & q_{23} \\ 0 & 0 & 0 & q_{13} & q_{23} & q_{33} \end{bmatrix} \tag{4.3}$$

where σ_a is the standard deviation related to target acceleration uncertainty, and

a definition of q_{ij} is [95]:

$$q_{11} = (1 - l + 2m + 2/3m^3 - 2m^2 - 4m\sqrt{l}) \quad (4.4)$$

$$q_{12} = \alpha(l + 1 - 1/l - 2m + m^2) \quad (4.5)$$

$$q_{13} = \alpha^2(1 - l - 2m\sqrt{l}) \quad (4.6)$$

$$q_{22} = \alpha^2(4\sqrt{l} - 3 - l + 2m) \quad (4.7)$$

$$q_{23} = \alpha^3(l + 1 - 2\sqrt{l}) \quad (4.8)$$

$$q_{33} = \alpha^4(1 - l) \quad (4.9)$$

where $l = e^{-2\alpha T_s}$ and $m = \alpha T_s$.

4.2 Sensor Model

This study assumes that the UAV is equipped with a VMTI or GMTI sensor to localise the position of the target. Using the pixel location of the target in an image, measurements of UAV position and attitude together with the camera pose angles and the camera model, the target is localised in world coordinates through corresponding geometric transformations as detailed in [97]. However, as our data provided by Roke Manor Research Ltd [2] for this study do not contain above information regarding vision process, precise modelling of the VMTI sensor is not possible in this case. Hence this study considers two simplified sensor models. The first one is a linear sensor model which uses the x and y positions of a ground vehicle as measurements, which can be modelled as:

$$\mathbf{z}_k = H_k \mathbf{x}_k^t + \nu_k \quad (4.10)$$

where the measurement matrix is $H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$. The covariance of measurement noise ν_k can be expressed as:

$$R_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (4.11)$$

where σ_x and σ_y are the standard deviations of the position of x and y, respectively. Since the current VMTI measurements from Roke Manor are target positions, the

measurement equation can be used directly for state estimation of ground vehicles, as detailed in section 4.3.

The second sensor model is based on a GMTI (Ground Moving Target Indicator) sensor which captures the characteristic and nonlinearity of the VMTI sensor more closely compared to the linear model. The measurements of relative range, azimuth, and the elevation angle with respect to the position of the UAV in a spherical coordinate are generated as follows:

$$\begin{aligned} r &= \sqrt{(x_k^t - x_k)^2 + (y_k^t - y_k)^2 + (z_k^t - z_k)^2} \\ \theta &= \tan^{-1} \frac{z_k^t - z_k}{\sqrt{(x_k^t - x_k)^2 + (y_k^t - y_k)^2}} \\ \phi &= \tan^{-1} \frac{y_k^t - y_k}{x_k^t - x_k} \end{aligned} \tag{4.12}$$

where $[M_x, M_y, M_z]$ and $[x_{uav}, y_{uav}, z_{uav}]$ are the positions of the ground vehicle and the UAV, respectively. Here, for simplification, M_z is assumed to be zero since the movement of ground vehicles can be assumed to be limited to two dimensions. In addition, UAVs are assumed to be flying at a constant altitude. The measurements in Cartesian coordinates are generated from the measurements in spherical coordinates, shown in Fig. 4.3, as:

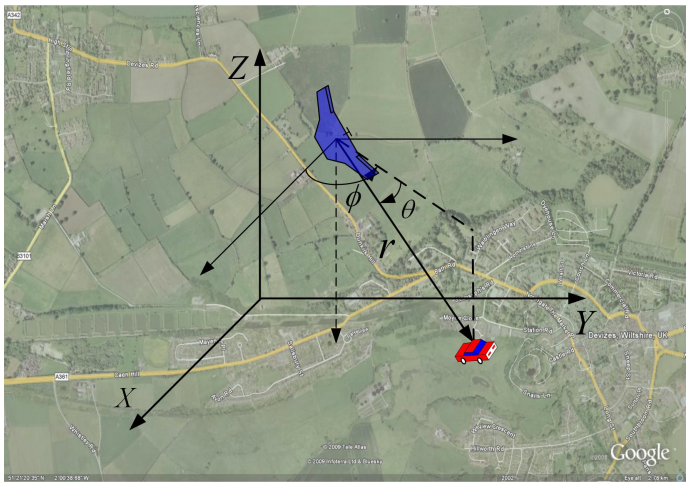


Figure 4.3: The relation between Cartesian and spherical coordinates

$$\begin{aligned} M_x &= r\cos\theta\cos\phi \\ M_y &= r\cos\theta\sin\phi. \end{aligned} \tag{4.13}$$

Once the measurements are passed through this transformation, they are in a suitable form to be input to the measurement equation in Eq. (4.10). In addition, the measurement covariance matrix is computed at every sampling time by considering the transformation from spherical to Cartesian coordinates as follows [95]:

$$R_k = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{4.14}$$

where

$$\begin{aligned} r_{11} &= \sigma_r^2 \cos^2\phi^2 r^2(\sigma_\phi^2 \sin\phi^2 \cos\theta^2 + \sigma_\phi^2 \cos\phi^2 \sin\phi^2) \\ r_{22} &= \sigma_r^2 \cos^2\phi^2 \sin\phi^2 r^2(\sigma_\phi^2 \sin\phi^2 \sin\phi^2 + \sigma_\phi^2 \cos\phi^2 \cos\phi^2) \\ r_{12} &= 0.5(\sigma_r^2 \cos^2\phi^2 \sin 2\phi + r^2(\sigma_\phi^2 \sin\phi^2 \sin 2\phi - \sigma_\theta^2 \cos\phi^2 \sin 2\phi)) \\ r_{21} &= r_{12}. \end{aligned} \tag{4.15}$$

Here, σ_r, σ_φ, and σ_θ are the standard deviations of the range, elevation, and azimuth respectively. Since real VMTI measurements are the positions of the target rather than relative range, azimuth and elevation, this sensor model will be used in conjunction with the virtually generated measurement data (mainly from S-Paramics traffic simulator) to support the proposed algorithms throughout this study.

If we consider only two-dimension (2-D) with range and azimuth of the target, sensor model can be simplified using the target position $(x_k^t, y_k^t)^T$ and the UAV position $(x_k, y_k)^T$ as nonlinear equation:

$$\mathbf{z}_k = \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} = h(\mathbf{x}_k^t) + v_k = \begin{bmatrix} \sqrt{(x_k^t - x_k)^2 + (y_k^t - y_k)^2} \\ \tan^{-1}\frac{y_k^t - y_k}{x_k^t - x_k} \end{bmatrix} + v_k \tag{4.16}$$

where $\tan^{-1}$ is to be executed as the four quadrant inverse tangent in practice, v_k is a measurement noise vector, and its noise covariance matrix is defined as:

$$V[v_k] = R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix}. \tag{4.17}$$

4.3 Conventional Target Tracking Filter

4.3.1 Kalman Filter

To estimate the states (position, velocity and acceleration) of a ground vehicle using the target dynamics and sensor model required for behaviour recognition, a Kalman filter (KF) is designed [98]. For a general discrete KF, several steps are required. Firstly, extrapolation of the states and the associated error covariance matrix is performed using the target dynamic model in the form:

Prediction

$$\mathbf{x}^t_{k|k-1} = F_k \mathbf{x}^t_{k-1|k-1} \tag{4.18}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k. \tag{4.19}$$

Next a correction step is performed which produces a minimum variance estimate of the states with an associated error covariance matrix, using measurements and a measurement model using equation Eq. (4.10).

Measurement update

$$K = P_{k|k-1} H_k^T \{H_k P_{k|k-1} H_k^T + R_k\}^{-1} \tag{4.20}$$

$$\mathbf{x}^t_{k|k} = \mathbf{x}^t_{k|k-1} + K\{\mathbf{z}_k - H_k \mathbf{x}^t_{k|k-1}\} \tag{4.21}$$

$$P_{k|k} = (I - KH_k) P_{k|k-1} \tag{4.22}$$

where $\mathbf{z}_k$ is the measurement vector and K is the Kalman gain.

4.3.2 Extended Kalman Filter

Considering $h(\mathbf{x}^t_k)$ in the measurement equation is nonlinear as shown in Eq. (4.16), the localization of target can be designed using the EKF (Extend Kalman Filter) with the same prediction equations in the above KF as [98]:

Measurement update

$$K = P_{k|k-1}H_k^T\{H_kP_{k|k-1}H_k^T + R_k\}^{-1} \quad (4.23)$$
$$\mathbf{x}_{k|k}^t = \mathbf{x}_{k|k-1}^t + K\{\mathbf{z}_k - h(\mathbf{x}_{k|k-1}^t)\} \quad (4.24)$$
$$P_{k|k} = (I - KH_k)P_{k|k-1}. \quad (4.25)$$

The output matrix H_k is a Jacobian of Eq. (4.16) with respect to the time-update state $\mathbf{x}_{k|k-1}^t$ as

$$H_k = \frac{\partial h}{\partial \mathbf{x}_k^t}|_{\mathbf{x}_k^t = \mathbf{x}_{k|k-1}^t} \quad (4.26)$$

where

$$\frac{\partial h}{\partial \mathbf{x}_k^t} = \begin{bmatrix} \frac{x_k^t - x_k}{\sqrt{(x_k^t-x_k)^2+(y_k^t-y_k)^2}} & 0 & 0 & \frac{y_k^t - y_k}{\sqrt{(x_k^t-x_k)^2+(y_k^t-y_k)^2}} & 0 & 0 \\ -\cos^2\{\tan^{-1}(\frac{y_k^t-y_k}{x_k^t-x_k})\}\frac{y_k^t-y_k}{(x_k^t-x_k)^2} & 0 & 0 & \cos^2\{\tan^{-1}(\frac{y_k^t-y_k}{x_k^t-x_k})\}\frac{1}{x_k^t-x_k} & 0 & 0 \end{bmatrix}. \quad (4.27)$$

4.3.3 Extended Information Filter

Considering the advantage of information form in multisensor systems such as easy initialisation, less communication load and dimension of measurements instead of states, the localisation of a target can also be designed by using the extended Information filter [99] as:

Prediction

$$\mathbf{y}_{k|k-1}^t = Y_{k|k-1}F_kY_{k-1|k-1}^{-1}\mathbf{y}_{k-1|k-1}^t \quad (4.28)$$
$$Y_{k|k-1} = (F_kY_{k-1|k-1}^{-1}F_k^T + Q_k)^{-1} \quad (4.29)$$

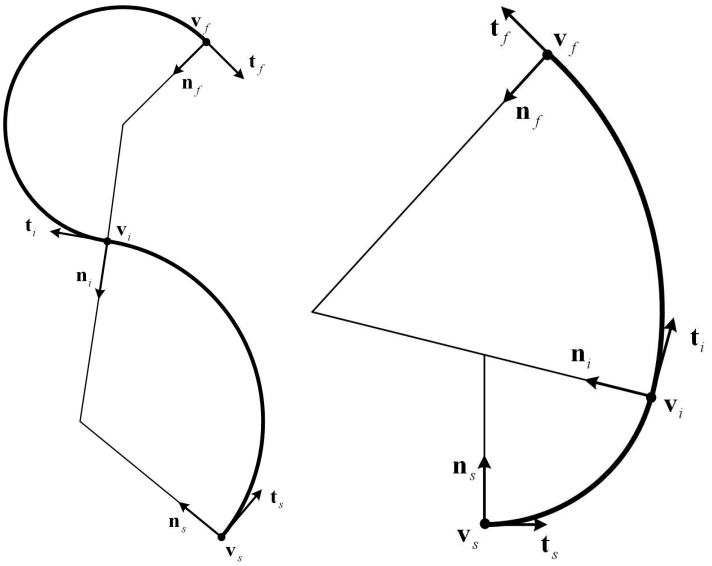
Measurement update

$$\mathbf{y}_{k|k}^t = \mathbf{y}_{k|k-1}^t + H_k^T(R_k)^{-1}[\mathbf{z}_k - h_a(\mathbf{x}_{k|k-1}^t) + H_k\mathbf{x}_{k|k-1}^t] \quad (4.30)$$
$$Y_{k|k} = Y_{k|k-1} + H_k^T(R_k)^{-1}H_k \quad (4.31)$$

where $Y_k = (P_k)^{-1}$ and $\mathbf{y}_k^t = Y_k\mathbf{x}_k^t$ represent the information matrix and information state vector, respectively.

4.4 Road-Constrained Filter

To make use of road-map information for the estimation of a target travelling on a road, it is required to express the road-map as a certain type of mathematical function which approximates the real road obtained from a given geographical database. To do so, this study uses constant curvature segments as used in Section 2.2. In this approach, assuming that some vertices on the road can be obtained, those vertices are connected by line segments of constant curvature with C^1 contact. The curved line between the vertices is to represent the curved nature of the real road. To ensure the C^1 constraints at both positions, an intermediate vertex is introduced such that the line segment is replaced by two arc segments of different curvature as shown in Fig. 4.4, as described in Section 2.2.



(a) Orientation 1 (b) Orientation 2

Figure 4.4: Arc segments with C^1 contact and intermediate vertex

The entire road-map can then be modelled by a set of road segments $r_i, i \in \{1, \ldots, n_r\}$, and for each road segment, the centre position of the curve and its curvature are given by the approximation algorithm. Figure 4.5 illustrates the road approximation using UAV sensor and constant curvature segments. As the UAV acquires some of points on the road from the vision or other sensors (marked as a cross in Fig. 4.5), the road is generated and extended successively. If a new point lies on or around the tangent line of a previous point, the road can be approximated as a straight line. Especially, in case that road information is

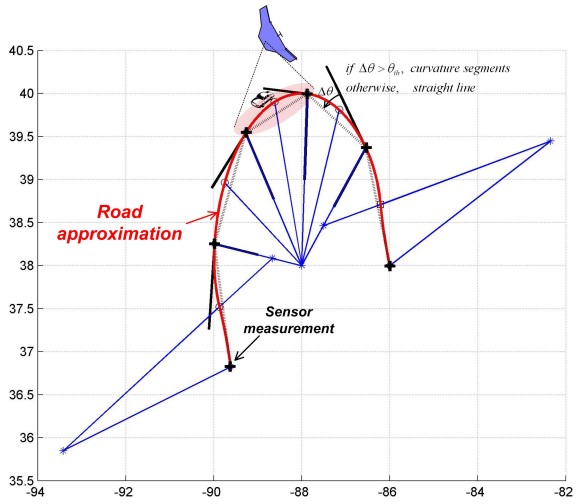


Figure 4.5: An illustration of road approximation using UAV sensor

not given in some area, this approach can be of interest with its efficiency since only some of the points on the road and corresponding segment curvature by the algorithm are required to approximate roads quite close to real roads. This can be readily exploited for the precise estimation of succeeding ground target on the road. The mathematical details of the construction of the line curvature between vertices are described in Section 2.2 and [10].

Figure 4.6 shows a sample road-network of Devizes, Wiltshire, United Kingdom, together with GIS (Geographic Information System) satellite data. Information for the road of interest, represented as the blue line, is assumed to be known in this study. Figs. 4.7~4.8 shows the approximated road and curvature for each road segment using some of known points on the road. Apparently, the more vertices are used, the better the fit to the road. However, since too many road segments might cause performance degradation in the constrained estimation, the appropriate number of vertices on the road needs to be determined to get a reasonable fit considering the road-network structure.

Now, assuming that the ground vehicle moves along given road-map consisting of n road segments, the 2-D position of the vehicle should lie on the one of roads. This can be expressed as the following constraint:

$$r_i(x_k^t, y_k^t) = 0 \tag{4.32}$$

where $r_i(\cdot)$ denotes the i-th road segment which can be modeled as straight line,

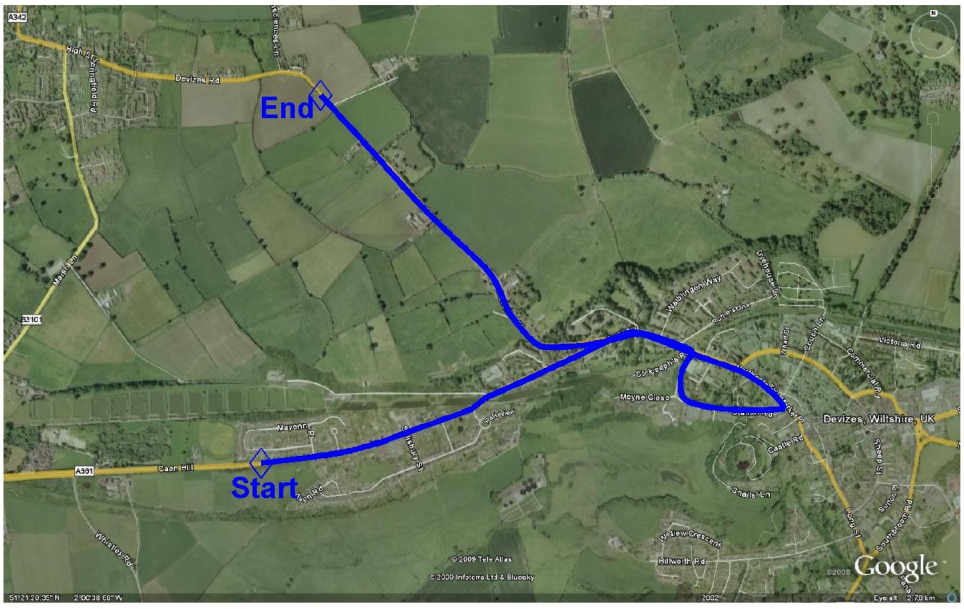
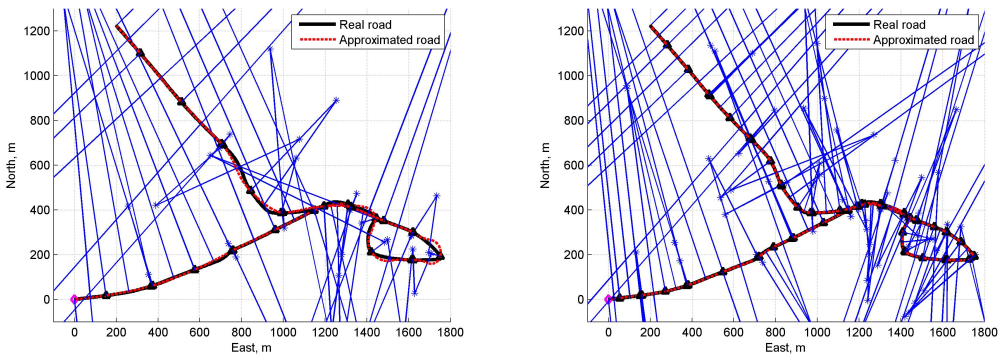


Figure 4.6: Sample road network with GIS satellite data overlaid (Google Map)

arc, or polynomials. For example, if the road is straight, the above road constraint can be expressed as:

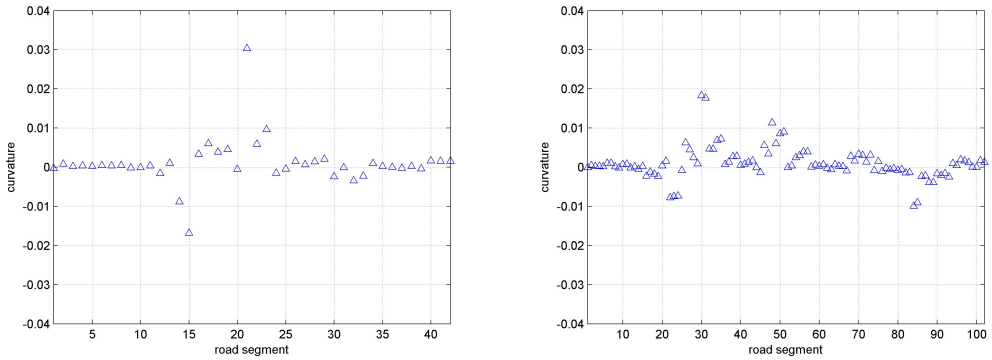
$$r_i(x_k^t, y_k^t) = \tan\theta \cdot x_k^t - y_k^t = 0 \tag{4.33}$$

where θ is a given road direction. In this study, since the road is approximated using constant curvature segments as explained earlier, the road constraint are



(a) Loosely represented roads *(b) Densely represented roads*

Figure 4.7: Road approximation using constant curvature segments



(a) Loosely represented roads (b) Densely represented roads

Figure 4.8: Curvature of each road segment

obtained as:

$$r_i(x_k^t, y_k^t) = (x_k^t - x_{i,ct})^2 + (y_k^t - y_{i,ct})^2 - \left(\frac{1}{\kappa_i}\right)^2 = 0 \tag{4.34}$$

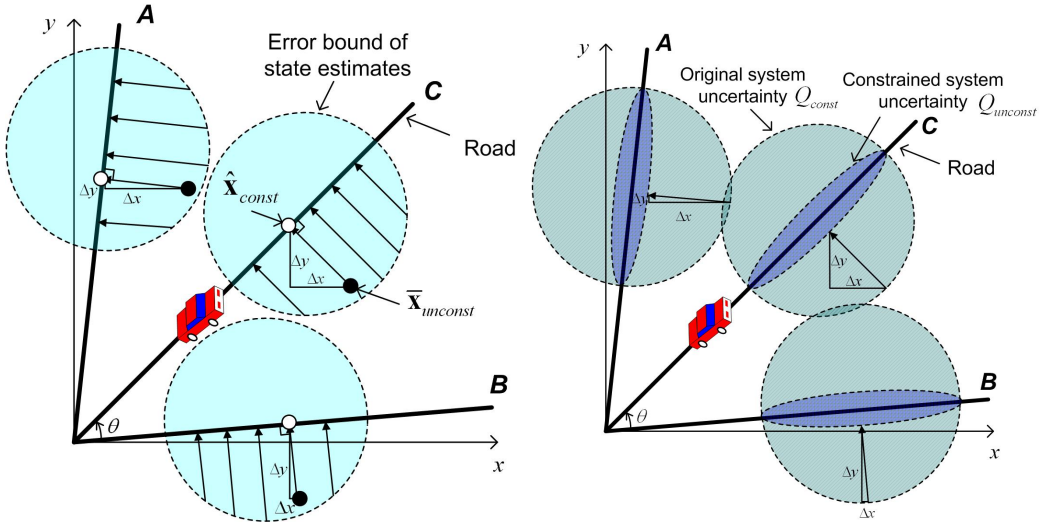
where $(x_{i,ct}, y_{i,ct})$ and κ_i are the centre position and the curvature of the i^{th} road segment, respectively.

Typically, there are two ways to deal with the road constraint in Kalman filtering framework. One is to use the road as equality constraints [88], and the other is to use the concept of a directional process noise [90], which represents uncertainty components along and orthogonal to the road, as illustrated in Fig 4.9. This study uses a pseudo-measurement method, one of the constrained Kalman filtering algorithms, which treats the equality constraints as additional fictitious or pseudo-measurement [89]. Unlike other approaches, such as the maximum probability method and the projection method [87], this approach has the advantage to enable consideration of the degree of constraint adherence, by monitoring the magnitude of the additional pseudo-measurement noise variance. The pseudo-measurement model using road constraints can be written as:

$$z_k^{r_i} = h_{r_i}(\mathbf{x}_k^t) + v_k^{r_i} \tag{4.35}$$

where $z_k^{r_i} = 0$, $h_{r_i}(\mathbf{x}_k^t) = r_i(\mathbf{x}_k^t)$, and $v_k^{r_i}$ is assumed to be a zero mean white Gaussian noise with covariance $R_k^{r_i} = (\sigma_r^{road})^2$ which accounts for the uncertainty of road constraints. Then, the previous real measurement model Eq. (4.16) is augmented by adding the pseudo-measurement to give:

$$\mathbf{z}_k^a = h_a(\mathbf{x}_k^t) + v_k^a \tag{4.36}$$



(a) Constrained Kalman filter (b) Directional process noise

Figure 4.9: Concept of road-constraint filtering

where $\mathbf{z}_k^a = [\mathbf{z}_k \; z_k^{r_i}]^T$, $h_a(\mathbf{x}_k^t) = [h(\mathbf{x}_k^t) \; h_{r_i}(\mathbf{x}_k^t)]^T$, and $v_k^a = [v_k \; v_k^{r_i}]^T$. The measurement noise covariance is also augmented to be $R_k^a = diag(R_k, R_k^{r_i})$. Considering that $h_a(\mathbf{x}_k^t)$ is nonlinear, the localisation of a target can be done by using the EKF with the augmented measurement equation, which will be called as the measurement-augmented EKF (MAEKF), in the form:

Measurement update

$$\begin{aligned} v_k &= \mathbf{z}_k^a - h_a(\mathbf{x}_{k|k-1}^t) && (4.37)\\ S_k &= H_k P_{k|k-1} H_k^T + R_k^a && (4.38)\\ \mathbf{x}_{k|k}^t &= \mathbf{x}_{k|k-1}^t + P_{k|k-1} H_k^T S_k^{-1} v_k && (4.39)\\ P_{k|k} &= (I - P_{k|k-1} H_k^T S_k^{-1} H_k) P_{k|k-1}. && (4.40) \end{aligned}$$

The output matrix H_k is a Jacobian of h_a with respect to the time-update state $\mathbf{x}_{k|k-1}^t$. Note that the EIF also can be applied with augmented measurement equation using the same way above. As a target is moving from one road segment to another, an appropriate road on which the target is travelling is selected, based on its estimated or a priori target position, its error covariance and the road network information as:

$$\begin{bmatrix} x_e^{r_i} - x_{k|k-1}^t \\ y_e^{r_i} - y_{k|k-1}^t \end{bmatrix}^T \left[P_{k|k-1}^{pos}\right]^{-1} \begin{bmatrix} x_e^{r_i} - x_{k|k-1}^t \\ y_e^{r_i} - y_{k|k-1}^t \end{bmatrix} < \varepsilon \qquad (4.41)$$

where $(x_e^{r_i}, y_e^{r_i})$ is the end position of i-th road segment, ε is the gate threshold parameter, and $P^{pos}_{k|k-1}$ is the position submatrix of the prediction covariance $P_{k|k-1}$. Here, a current road segment is updated to the next segment once the above condition is satisfied.

In case that nonlinearity of the road segment is severe, since the EKF based on linearisation can result in poor performance, this study also designed the UKF, and compares the results between those two filtering methods. Road constraints can be incorporated into the UKF by treating it as a pseudo-measurement with the similar way as in the EKF but without liniearisation of constraints being able to provide better accuracy. The UKF is a filter for nonlinear systems which uses sigma points approximating a given PDF [100]. Among various UKF methods dealing with pseudo-measurements, an equality constrained UKF (ECUKF) is adopted in this study considering its reasonable performance and computation time [101]. In the ECUKF, at each update step, the stated estimate of the unconstrained UKF is combined with the constraints, which are treated as pseudo-measurement, to obtain a constrained a posteriori UKF estimate.

4.5 Optimal Fixed-Interval Smoother

In Section 4.3, a discrete Kalman filter tracker for a ground vehicle was designed using the acceleration model [95] with an assumed level of sensor noise in spherical coordinates. Although it can provide reasonable estimates for the ground vehicles, the estimation error is not often small enough to capture local-level manoeuvres of a ground vehicle, such as lane-changing. Given that roadmap information is not available, the recent past history of the state estimates can be exploited to enhance the tracking accuracy since the behaviour classification for the ground vehicle we consider in Section 5 uses the trajectory history over a specific length of time rather than instantaneously. For this case, an optimal fixed-interval smoother [102] can be combined with the Kalman filter.

This smoothing algorithm is composed of a forward filter and a backward filter as shown in Fig. 4.10. The basic idea is that, if the measurements between t and $T_f (> t)$ are available, the estimates of the forward filter at time t, $\hat{x}_f(t)$, can be adjusted, based on the estimates of the backward filter at that time, $\hat{x}_b(t)$.

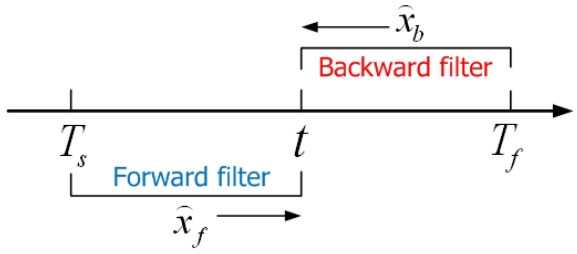


Figure 4.10: A concept of optimal smoother

From now on, superscripts f and b designate the forward filter and the backward filter, respectively. The detail derivation of the optimal smoother can be found in [98, 103], and briefly explained as follows.

The optimal fixed interval smoothing requires three passes through the measurements and data derived therefrom, at every discrete time t_k in the entire fixed interval:

1. A complete filter pass in the forward direction (i.e., with measurement time increasing), saving the values of the *a priori estimates* $\hat{x}_k^t(+)$ and the associated covariance of estimation uncertainty $\hat{P}_k^f(+)$ (which are equivalent to those in Eqs. (4.21)~(4.22)).

2. A complete filter pass in the backward direction (time decreasing), saving the *a priori estimates* and associated covariance of estimation uncertainties. Let us assume the following variable transformation.

$$S_k(\pm) \triangleq P_k^b(\pm)^{-1} \tag{4.42}$$
$$\hat{y}_k(\pm) \triangleq P_k^b(\pm)^{-1}\hat{x}_k^b(\pm) = S_k(\pm)\hat{x}_k^b(\pm) \tag{4.43}$$

If the last discrete sampling time is N, the transformed estimates and error covariance matrix can be initialized using:

$$\hat{y}_N(-) = 0 \tag{4.44}$$
$$S_N(-) = 0. \tag{4.45}$$

The backward filter performs the measurement update first:

$$\hat{y}_k(+) = \hat{y}_k(-) + H_k^T R_k^{-1}\mathbf{z}_k \tag{4.46}$$
$$S_k(+) = S_k(-) + H_k^T R_k^{-1} H_k \tag{4.47}$$

Then, the time update is performed using the following equations.

$$
\begin{aligned}
K_k^b &= S_k(+)(S_k(+) + Q_k^{-1})^{-1} && (4.48)\\
S_{k-1}(-) &= F_k^T(I - K_k^b)S_k(+)F_k && (4.49)\\
\hat{y}_{k-1}(-) &= F_k^T(I - K_k^b)\hat{y}_k(+) && (4.50)
\end{aligned}
$$

3. A third, smoother pass combining the forward and backward data to obtain the smoothed estimate and the covariance of smoother uncertainty,

$$
\begin{aligned}
K_k &= P_k^f(+)S_k(-)(I + P_k^f(+)S_k(-))^{-1}\\
P_k &= (I - K_k)P_k^f(+)\\
\hat{x}_k &= (I - K_k)\hat{x}_k^f(+) + P_k\hat{y}_k(-). && (4.51)
\end{aligned}
$$

4.6 Sensor Fusion Techniques

For the case where multiple UAVs equipped with a VMTI or GMTI sensor are involved, sensor fusion techniques can be applied to optimally fuse the individual estimates. In this section, the concept of sensor fusion is first explained. Then, some of widely-used sensor fusion algorithms (covariance intersection, state-vector fusion, and decentralised fusion) in line with the target estimation amongst many others such as Bayesian reasoning, fuzzy logic, and neural networks [104] are successively described.

4.6.1 Introduction to Sensor Fusion

Sensor fusion is a well-known technique where multiple sources of data are fused together in order to get a better estimate of the underlying information. The input sources are almost always corrupted by noise, and two such signals can be represented by random variables a and b. The real statistics of these variables a and b are assumed to be unknown. They, in turn, provide the only information for consistent estimates of the means and the covariances of these variables. In order to define the consistency of these two variables let the means of these variables be designated as $\bar{a}$ and $\bar{b}$ and the deviations around the means are designated as

$\tilde{a} \triangleq a - \bar{a}$ and $\tilde{b} \triangleq b - \bar{b}$. In general, these deviations are not zero mean, so that the mean square error and the cross-correlations can be derived as:

$$\bar{P}_{aa} = E[\tilde{a}\tilde{a}^T] \tag{4.52a}$$

$$\bar{P}_{bb} = E[\tilde{b}\tilde{b}^T] \tag{4.52b}$$

$$\bar{P}_{ab} = E[\tilde{a}\tilde{b}^T]. \tag{4.52c}$$

Since the true values are unknown, it has been approximated by the values of P_{aa} and P_{bb}. So, these approximations are only consistent if it satisfies the following conditions.

$$P_{aa} - \bar{P}_{aa} \geqq 0 \tag{4.53}$$

$$P_{bb} - \bar{P}_{bb} \geqq 0 \tag{4.54}$$

The signals given by variables a and b are combined together in order to get a new estimate $\{\bar{c}, P_{cc}\}$,

$$P_{cc} - \bar{P}_{cc} \geqq 0 \tag{4.55}$$

where $\tilde{c} \triangleq c - \bar{c}$ and $\bar{P}_{cc} = E[\tilde{c}\tilde{c}^T]$.

If the statistics of these variables are known, the sensor fusion algorithm can be optimally addressed by computing a linear combination of the means of these variables, and then analytically determine the covariance of the result. This approach leads to the problem when there is uncertainty in the cross correlation with the unknown statistics. For instance, the Kalman filter uses a linear combination of the signals in the form of $\bar{c} = W_a\bar{a} + W_b\bar{b}$ and the resulting covariance is given as:

$$P_{cc} = W_a(P_{aa}W_a^T + P_{ab}W_b^T) + W_b(P_{ba}W_a^T + P_{bb}W_b^T). \tag{4.56}$$

The trace of P_{cc} is minimised with respect to W_a and W_b. Note that this calculation is done using the assumed covariance, however the actual covariance is:

$$\bar{P}_{cc} = W_a(\bar{P}_{aa}W_a^T + \bar{P}_{ab}W_b^T) + W_b(\bar{P}_{ba}W_a^T + \bar{P}_{bb}W_b^T). \tag{4.57}$$

If the assumed and the actual variables are uncorrelated ($P_{ab} = \bar{P}_{ab} = 0$) then a consistent update is possible. However, if $\bar{P}_{ab} \neq 0$ then it is hard to generate a consistent update. This problem led to the development of the Covariance Intersection method [100]. This is a sensor fusion algorithm having a convex combination of the means and the covariance in the information space which is briefly explained in the following section.

4.6.2 Covariance Intersection

The basic intuition behind the covariance intersection (CI) is to form a geometric interpretation of Eq. (4.56). When P_{cc} lies within the intersection of P_{aa} and P_{bb} for any possible choices of P_{ab}, then an update strategy in the intersection region P_{cc} will be consistent. The tighter the updated covariance fits the region of intersection, the more information is extracted. This intersection is characterised by a convex combination of the covariances and this CI algorithm is formulated by:

$$P_{cc}^{-1} = \omega P_{aa}^{-1} + (1-\omega)P_{bb}^{-1} \tag{4.58}$$

$$P_{cc}^{-1}\bar{c} = \omega P_{aa}^{-1}\bar{a} + (1-\omega)P_{bb}^{-1}\bar{b} \tag{4.59}$$

where $\omega \in [0,1]$. The free parameter ω defines the convex weights which are assigned to a and b. Different choices of ω can be used to optimise the covariance update with respect to different performance criteria, e.g. minimising the trace or the determinant of P_{cc}. The cost functions are convex with respect to ω and will have a unique optimum in the range of $0 \leq \omega \leq 1$. This CI algorithm is described in more detail in references [100] and [105].

The CI algorithm for two data sources can be extended to the general case having an arbitrary number of estimates [106], so that:

$$P_{cc}^{-1} = \omega_1 P_{a_1a_1}^{-1} + \omega_2 P_{a_2a_2}^{-1} + \cdots + \omega_n P_{a_na_n}^{-1} \tag{4.60}$$

$$P_{cc}^{-1}\bar{c} = \omega_1 P_{a_1a_1}^{-1}\bar{a}_1 + \omega_2 P_{a_2a_2}^{-1}\bar{a}_2 + \cdots + \omega_n P_{a_na_n}^{-1}\bar{a}_n \tag{4.61}$$

where $\sum_{i=1}^{n} \omega_i = 1$.

4.6.3 State-Vector Fusion

State-vector fusion, also known as a track-to-track fusion [107], can also be applied under the assumption that the communication bandwidth is wide enough to transmit the state (6-by-1) and covariance matrix (6-by-6) in both directions between pairs of UAVs. At each sampling step, the combined target state and covariance matrix is given by:

$$\hat{\mathbf{x}}_k^t = \mathbf{x}_{k|k}^t + P_{k|k}(P_{k|k} + P_{k|k}^p)^{-1}(\mathbf{x}_{k|k}^{tp} - \mathbf{x}_{k|k}^t) \tag{4.62}$$

$$P_k = P_{k|k} + P_{k|k}(P_{k|k} + P_{k|k}^p)^{-1}P_{k|k}^T \tag{4.63}$$

where $\mathbf{x}_{k|k}^{tp}$ and $P_{k|k}^{p}$ represent the state and error covariance estimations of the UAV pair.

4.6.4 Decentralised Sensor Fusion

Coordination and sharing of information amongst UAVs can provide a better understanding of the environment as well as producing higher survivability of the group in a given mission. Since it is almost infeasible to have a single centralised agent that handles the entire information processing and the performance of tasks for making a decision, given the real-world communication and computation constraints, distributed and decentralised approaches have been developed in both the estimation and cooperative control area [99, 108] with good scalability and robustness properties.

Given the fact that multiple UAVs will carry out the process of tracking ground targets, each UAV will obtains its own sensor measurement and executes the tracking filter algorithm separately. After each UAV receives the other's estimation via a communication link, it can run a decentralised sensor fusion or consensus of information process to enhance the tracking accuracy. Information exchange process between agents is typically modelled by directed or undirected graphs [108]. Suppose that there are n UAVs $U = \{U_1, ..., U_n\}$ with inconsistent information of the target. The communication pattern at time step k can be expressed as a directed graph $\mathbb{G}_k = (U, \varepsilon_k)$, where $(U_i, U_j) \in \varepsilon^k$, if and only if there is a unidirectional information exchange link from U_i to U_j. The adjacency matrix $G_k = [g_k^{ij}]$ of a graph $\mathbb{G}_k$ is defined as:

$$g_k^{ij} = \begin{cases} 1 & \text{if} \quad (U_j, U_i) \in \varepsilon_k \\ 0 & \text{if} \quad (U_j, U_i) \notin \varepsilon_k \end{cases} \tag{4.64}$$

Note that $g_k^{ii} \triangleq 1$ for a link from each UAV to itself. From the following, two different decentralised estimation algorithms are presented.

4.6.4.1 Decentralised extended information filter

Decentralised estimation is first performed by decentralising the extended Information filter algorithm to produce the decentralised extended Information filter

(DEIF) [99]. Each UAV computes local predictions based on previous, locally determined information estimates using its system model, in the form:

Prediction

$$\mathbf{y}^t_{i,k|k-1} = Y_{i,k|k-1} F_k Y^{-1}_{i,k-1|k-1} \mathbf{y}^t_{i,k-1|k-1} \tag{4.65}$$

$$Y_{i,k|k-1} = (F_k Y^{-1}_{i,k-1|k-1} F^T_k + Q_k)^{-1} \tag{4.66}$$

where $Y_{i,k} = (P_{i,k})^{-1}$ and $\mathbf{y}^t_{i,k} = Y_{i,k}\mathbf{x}^t_{i,k}$ represent the information matrix and information state vector of the i^{th} UAV. The predictions are combined with information from local observations to compute local estimates which will then be communicated between UAVs over the communication network. After exchanging this data, information estimates for the i^{th} UAV are updated with information obtained from the other UAVs at time step k represented by $i_{j,k}$ and $I_{j,k}$ to give:

$$\mathbf{y}^t_{i,k|k} = \mathbf{y}^t_{i,k|k-1} + \sum_{j=1, j\neq i}^{N} g^{ij}_k i_{j,k} \tag{4.67}$$

$$Y_{i,k|k} = Y_{i,k|k-1} + \sum_{j=1, j\neq i}^{N} g^{ij}_k I_{j,k} \tag{4.68}$$

where

$$i_{j,k} = H^T_{j,k}(R_{j,k})^{-1}[\mathbf{z}_{j,k} - h_j(\mathbf{x}^t_{j,k|k-1}) + H_{j,k}\mathbf{x}^t_{j,k|k-1}] \tag{4.69}$$

$$I_{j,k} = H^T_{j,k}(R_{j,k})^{-1}H_{j,k} \tag{4.70}$$

and where subscript j is associated with information from j^{th} UAV.

4.6.4.2 Kalman consensus algorithm

This section briefly explains the Kalman consensus algorithm. Suppose $\mathbf{x}^t_{i,k}$ represents the information of UAV U_i about the target with the true value $\mathbf{x}^*$ at time step k. Assuming that this true value is constant, the system dynamics can be modelled as $\mathbf{x}^*_{k+1} = \mathbf{x}^*_k + w_k$ where $w_k \sim N(0, Q^c_k)$ is a zero-mean disturbance input. Letting

$v^c_{k,ij} \sim N(0, \Omega^{ij}_k)$ represent the communication noise, the measurement equation for UAV U_i which represents the information exchange can be written by:

$$\mathbf{z}^c_{i,k} = \begin{bmatrix} g^{i1}_k(\mathbf{x}^t_{1,k} + v^c_{k,i1}) \\ \vdots \\ g^{in}_k(\mathbf{x}^t_{n,k} + v^c_{k,in}) \end{bmatrix}.$$

Assuming that the UAV's estimation errors are uncorrelated, i.e., $E[(\mathbf{x}^t_{i,k} - \mathbf{x}^*_k)(\mathbf{x}^t_{j,k} - \mathbf{x}^*_k)^T] = 0$, where $i \neq j$, and by defining $P^c_{i,k} = E[(\mathbf{x}^t_{i,k} - \mathbf{x}^*_k)(\mathbf{x}^t_{i,k} - \mathbf{x}^*_k)^T]$, the discrete-time Kalman consensus algorithm for the i^{th} UAV can be written as [109]:

$$P^c_{i,k+1} = [(P^c_{i,k} + Q^c_k)^{-1} + \sum_{j=1, j\neq i}^{N} g^{ij}_k (P^c_{j,k} + \Omega^{ij}_k)^{-1}]^{-1} \tag{4.71}$$

$$\mathbf{x}^t_{i,k+1} = \mathbf{x}^t_{i,k} + P^c_{i,k+1} \sum_{j=1, j\neq i}^{N} [g^{ij}_k (P^c_{j,k-1} + \Omega^{ij}_k)^{-1} (\mathbf{x}^t_{j,k} + v^{k,ij}_c - \mathbf{x}^t_{i,k})]. \tag{4.72}$$

Compared to the DEIF, this algorithm has the advantage of being able to consider the communication noise systematically as the normal Kalman filtering does.

4.7 Numerical Simulations

This section carries out numerical simulations using the presented tracking filters for a moving ground vehicle. The ground target trajectory is obtained from S-Paramics [94] traffic model as shown in Fig 4.11 at 2 Hz. It is used to generate sensor measurements composed of relative range, elevation, and azimuth having the white noise of $(\sigma_r, \sigma_\varphi, \sigma_\theta) = (10\ m, 0.005\ rad, 0.005\ rad)$. The results of the optimal smoother are compared to those of the Kalman tracker to check the performance enhancement. Figures 4.12~4.13 show the true estimated trajectories with the loitering UAV and position estimation error in the x and y coordinates, respectively. Speed and acceleration estimations can be found in Figs. 4.14~4.15. It can be seen that the performance of the optimal smoother is more accurate than that of the Kalman filter. In particular, while the Kalman filter has a difficulty to track the true acceleration, the optimal smoother has a tendency to follow it more closely as shown in Fig. 4.15. Note that since the true acceleration values

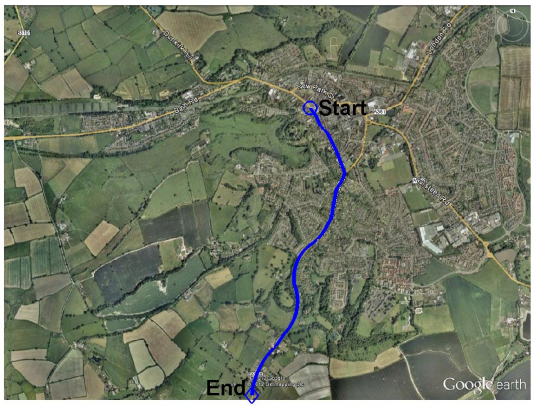


Figure 4.11: Trajectory of a ground vehicle in S-Paramics model with GIS satellite data overlaid thanks to Goolge earth

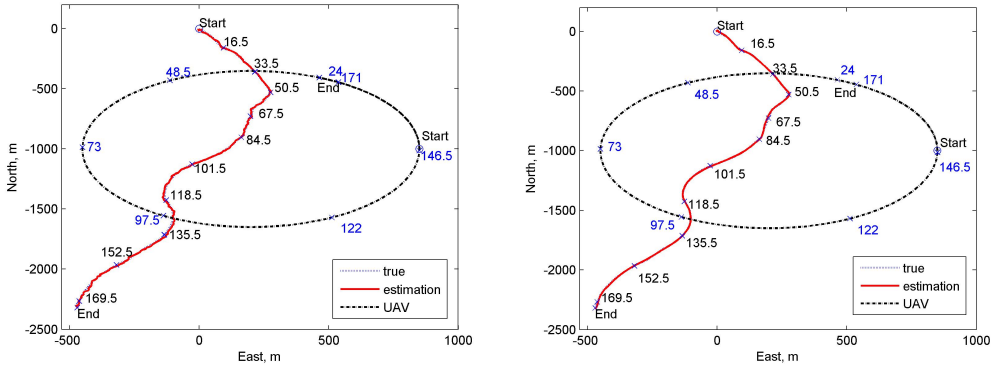


Figure 4.12: Trajectory estimations of a ground vehicle in S-Paramics Devizes model, the general Kalman tracker (left) and the optimal smoother (right)

are obtained by twice differentiation of true positions, they have unrealistic peak values (even greater than 10 m/s^2) frequently, which need to be ignored.

Besides, to verify the performance of the sensor fusion algorithms, we carried out the simulation under the cylindrical station-keeping of two UAVs flying with 100 km/h at the different altitudes, $h_1 = 500\ m$ and $h_2 = 800\ m$ as shown in Figure 4.16. The diameter of the cylinder, d, is 650 m, and its center locates on (200 m,-1000 m). UAV1 flies counter-clockwise, and UAV2 does clockwise. Figure 4.16 shows the trajectory of a ground vehicle by covariance intersection with optimal smoother using two UAVs. The process that a new covariance ellipse (green one) is computed with that of each UAV's optimal smoother using covariance intersection

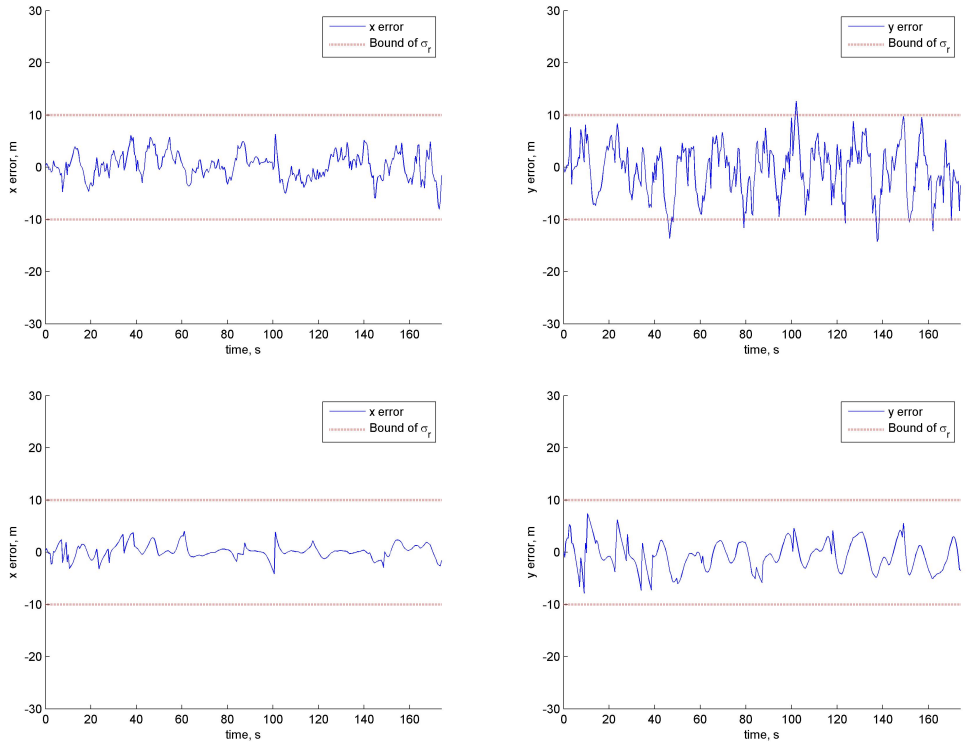


Figure 4.13: Position estimation error of the coordinate of a ground vehicle in S-Paramics Devizes model, the general Kalman tracker (top) and the optimal smoother (bottom)

algorithm is described in Fig. 4.17. Table 4.1 and 4.2 show the tracking accuracy enhanced by sensor fusion algorithms. It can be observed that state-vector fusion algorithm shows slightly better performance than covariance intersection.

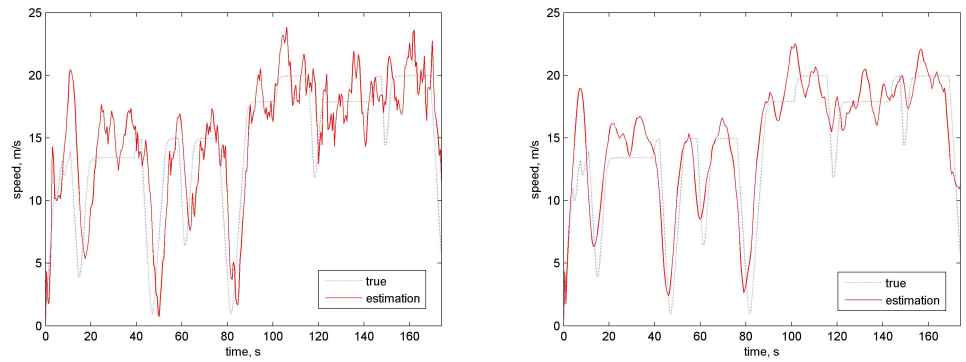


Figure 4.14: Speed estimation of a ground vehicle in S-Paramics Devizes model, the general Kalman tracker (left) and the optimal smoother (right)

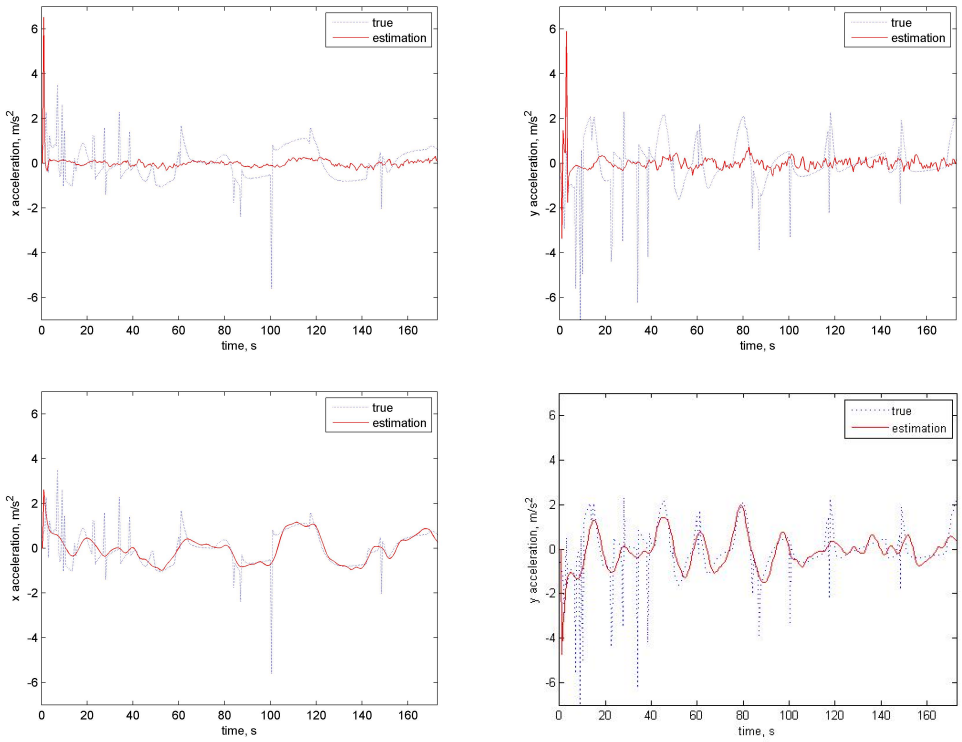


Figure 4.15: Acceleration estimation of a ground vehicle in S-Paramics Devizes model, the general Kalman tracker (top) and the optimal smoother (bottom)

4.8 Summary

This chapter presented ground moving target estimation algorithms for the UAV surveillance problem as a background of this thesis, which will be used in the following chapters. The conventional target tracking method is first explained using an acceleration model as a realistic target dynamics and simplified mea-

Table 4.1: Analysis of estimation error in x-coordinate

	min (m)	mean (m)	max (m)
(1) Forward Kalman tracker by UAV1	0.0284	2.8869	11.5003
(2) Forward Kalman tracker by UAV2	0.0118	2.7788	13.7168
(3) Optimal smoother by UAV1	0.0173	1.5669	5.2491
(4) Optimal smoother by UAV2	0.0025	1.3376	7.6801
(5) Covariance Intersection using (3) and (4)	0.0008	1.3413	4.8956
(6) State-vector fusion using (3) and (4)	0.0012	1.1838	5.4784

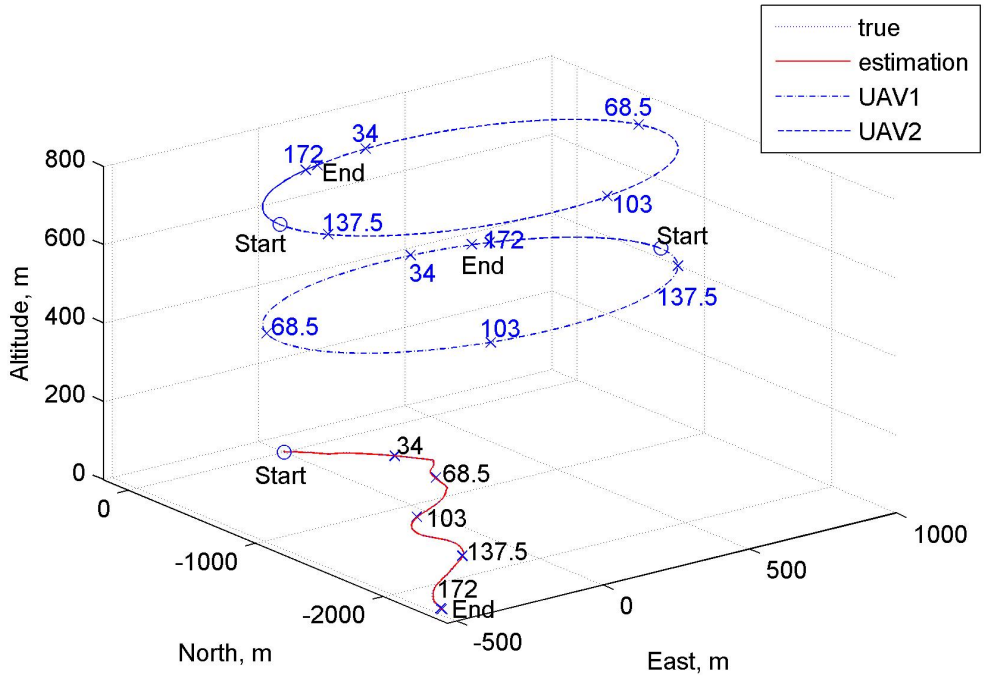


Figure 4.16: Trajectory of the ground vehicle by covariance intersection with the optimal smoother

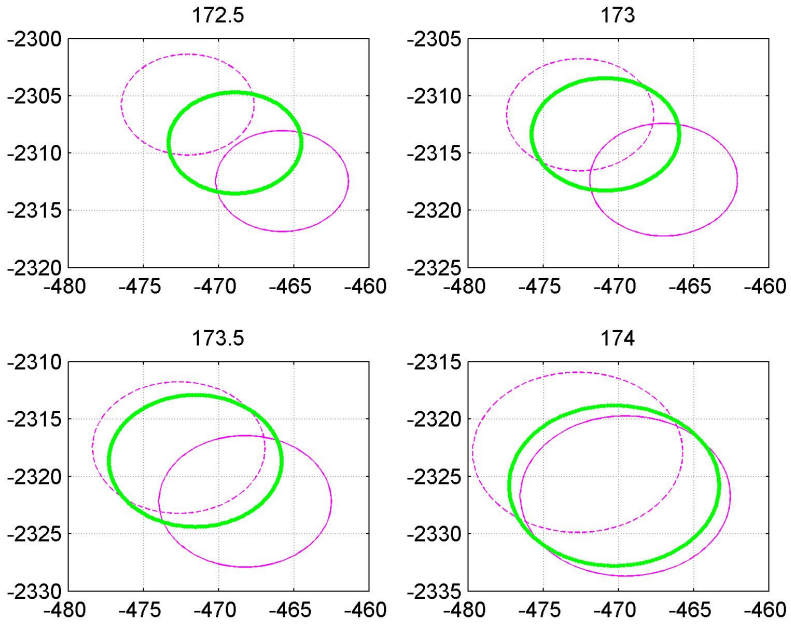


Figure 4.17: Covariance ellipses at the final four steps in case of the optimal smoother

Table 4.2: Analysis of estimation error in y-coordinate

	min (m)	mean (m)	max (m)
(1) Forward Kalman tracker by UAV1	0.0134	3.4055	19.4884
(2) Forward Kalman tracker by UAV2	0.0039	3.2312	19.1591
(3) Optimal smoother by UAV1	0.0091	1.7911	11.7327
(4) Optimal smoother by UAV2	0.0101	1.9339	18.5902
(5) Covariance Intersection using (3) and (4)	0.0012	1.7547	15.1614
(6) State-vector fusion using (3) and (4)	0.0020	1.6964	15.1614

surement models. Then, a practical road approximation algorithm and nonlinear road-constrained Kalman filtering are introduced, in order to exploit given road information for precise target estimation. Optimal smoothing and sensor fusion techniques were also presented to enhance the tracking accuracy further. Numerical simulation results using realistic car trajectory data from the traffic simulation program showed the performance and properties of estimation algorithms. Note that numerical simulations for the other filtering methods explained in this chapter are performed in the following chapters, especially in Section B and Chapter 6, in conjunction with anomaly detection and guidance algorithms.

Chapter 5

Airborne Surveillance and Monitoring of Ground Vehicle Behaviour

5.1 Introduction

Recently, airborne surveillance and monitoring systems have become a challenging and emerging issue in the area of aerospace and robotics, with the rapid improvement of UAV systems and of associated sensing technology. For instance, [15] analysed the feasibility of using multiple low-altitude, short endurance UAVs to cooperatively monitor and track the propagation of large forest fires. The interest for airborne surveillance was also extended to the roadway traffic monitoring [17]. Compared to traditional traffic surveillance sensors such as loop detectors and video cameras positioned at fixed locations, UAV aerial sensing can provide better coverage with the capability to survey large areas at a high speed without being confined to prescribed ground navigation routes [110].

In particular, UAV surveillance with onboard sensors can provide accurate information of a large number of moving targets and consequently allow rapid and consistent detection of possible threats. For this, a VMTI (Visual Moving Target Indicator) sensor in both visible and infrared bands has been developed recently to automatically detect and track moving targets on the ground [2]. When data from the VMTI sensor is combined with visual mosaic techniques, target positions and motions can be determined, as shown in Fig. 5.1, and hence can increase the operator's situational awareness significantly. However, the operator will still need to analyse the resulting position and motion data and construct a picture of events, in order to detect suspicious behaviour. This usually requires several highly-skilled human operators, which is expensive and unsustainable under a deluge of data and information from complex scenes. Hence, there is a strong need to develop high-level analysis algorithms to process target information and detect anomalous behaviours, to reduce the human operator's workload.

Activity modelling and automatic anomaly detection tasks are non-trivial since

Figure 5.1: Examples of VMTI operation: tracking and highlighting of moving objects (left) and visual mosaic (right) from website of Roke Manor Research Ltd [2]

complex activity patterns in a dynamic environment involve multiple objects interacting with each other over space and time, whilst anomalies are often rare, ambiguous and can be easily confused with sensor noise. In particular, anomalous behaviours of a ground vehicle are closely related to driving behaviours such as aggressive, drowsy, or intoxicated lane crossing manoeuvres [111, 112, 113, 114]. Also, location where a vehicle is moving, speed, and regional activity patterns (whose characteristics are often defined by the combination of location and speed) [115] are other important aspects to identify anomalous behaviours.

In general, detecting anomalous behaviour can be classified into two categories: The first approach codifies the behaviours using the experience and domain knowledge of experts, while in the second approach the behaviours are extracted and learned from the data [116]. Purely learning based approaches can provide good performance [115, 117, 118], however, they require massive data sets in advance, or tend to suffer from a high computation burden for real-time applications. On the other hand, there are several algorithms which deal with behaviour or activity analysis in the context of codified (or classified) behaviour models, with the aid of the learning approach, for both maritime and ground traffic surveillance scenarios.

In this regard, Srivastava *et al.* [119] introduced a method to detect anomalies of the ground vehicle motion by observing the patterns in its velocity trajectory using a hypothetical coordinate system. The shape of velocity trajectory is used to detect anomalies by reference to paths derived from a trained Gaussian mixture model. In [120], moving target trajectories are expressed as discrete pattern frag-

ments, known as motifs, along with several attributes such as location, duration of motif and average speed. With this motif-based feature space representation, hierarchical rule-based classifier are used to detect abnormal behaviours. Fraile and Maybank [121] proposed the idea of dividing the trajectories of the ground vehicles into distinct driving modes, using video images obtained for ground traffic surveillance. However, this classification is limited to slow speed car manoeuvres in an urban parking area.

This chapter proposes a behaviour recognition methodology against moving ground vehicles within road traffic using UAVs, in order to identify suspicious or abnormal behaviour, thereby reducing the workload of human operators. For this study, vehicular traffic modelling should be first addressed, which is a central problem of traffic engineering [122, 123]. This is generally classified into two main approaches differing in the level of detail: macroscopic and microscopic. Macroscopic models attempt to describe the traffic flow by interpreting the traffic flow as a compressible fluid without distinguishing between vehicles. In contrast to the macroscopic ones, microscopic models define the traffic behaviour by describing the behaviour of individual drivers in different situations. This study focuses on the microscopic-level differential geometric modelling since there is a definite need of tracking each individual vehicle's trajectory for irregular behaviour recognition.

Current research using UAVs has focused on the development of the separate algorithms, e.g. sensing and calibration [23, 24], multi-sensor management and information fusion [25, 26, 124], or guidance and control [27, 28] required for the ground traffic monitoring. However, they have not been integrated into a system to aid situation assessment. This study describes the work done to develop a systematic and integrated development for the autonomous recognition for airborne monitoring of ground traffic behaviour. It includes the following techniques: target tracking, sensor fusion and trajectory refinement, trajectory classification, and behaviour recognition. One of contributions of this study is to propose a classification methodology for driving behaviour represented as a sequence of integers used to categorise specific differential geometric quantities. Utilising this trajectory classification, a simple but effective detection scheme for irregular driving behaviour is proposed, based on string matching theory. In line with the suspicious behaviour detection scheme, an anomaly detection

algorithm is also proposed as can be found in Appendix B. This algorithm uses both a learning approach based on Gaussian processes and a domain knowledge approach provided from road map information, in order to produce a velocity profile of normal vehicles as well as detect deviations of the vehicle from the road. Another important contribution of this study is that real VMTI sensor measurements as well as realistic car simulation data, using an off-the-shelf traffic simulation program, is applied to effectively validate the performance of the integration system.

Although the aforementioned string matching and anomaly detection algorithms provide a measure of suspicious behaviour, additional information needs to be considered to finally confirm the characteristic of behaviour in order to avoid frequent false alarms. For instance, a geographical database or other domain knowledge describing the surrounding environments, such as traffic flow density or velocity data of normal vehicles could be of great help in efficiently identifying the intention of suspicious ground vehicles. Therefore, a fuzzy decision making process is developed to systematically exploit all available information obtained from a complex environment. The proposed fuzzy expert rule-based decision making system can concurrently accommodate several aspects of behaviour, as well as taking into account the spatiotemporal environment factors, thus providing a more effective level of alert to the operator monitoring complex scenes.

The structure of this chapter is given as follows. Section 5.2 explains behaviour modelling and recognition framework based on trajectory-based driving mode classification and string matching theory. Section 5.3 introduces a rule-based decision making to find suspicious or anomalous behaviour based on fuzzy logic. Lastly, conclusions and future work are addressed in Section 5.4. Figure 5.2 shows a flow chart of overall algorithm for the proposed automatic behaviour recognition scheme. Note that this figure uses the same structure as Fig. 1.2 in Chapter 1, while focusing on airborne monitoring and behaviour recognition.

5.2 Behaviour Modelling and Recognition

This section introduces behaviour modelling and detection algorithm of ground vehicles. Ground target tracking filter using UAVs is firstly explained. Trajectory

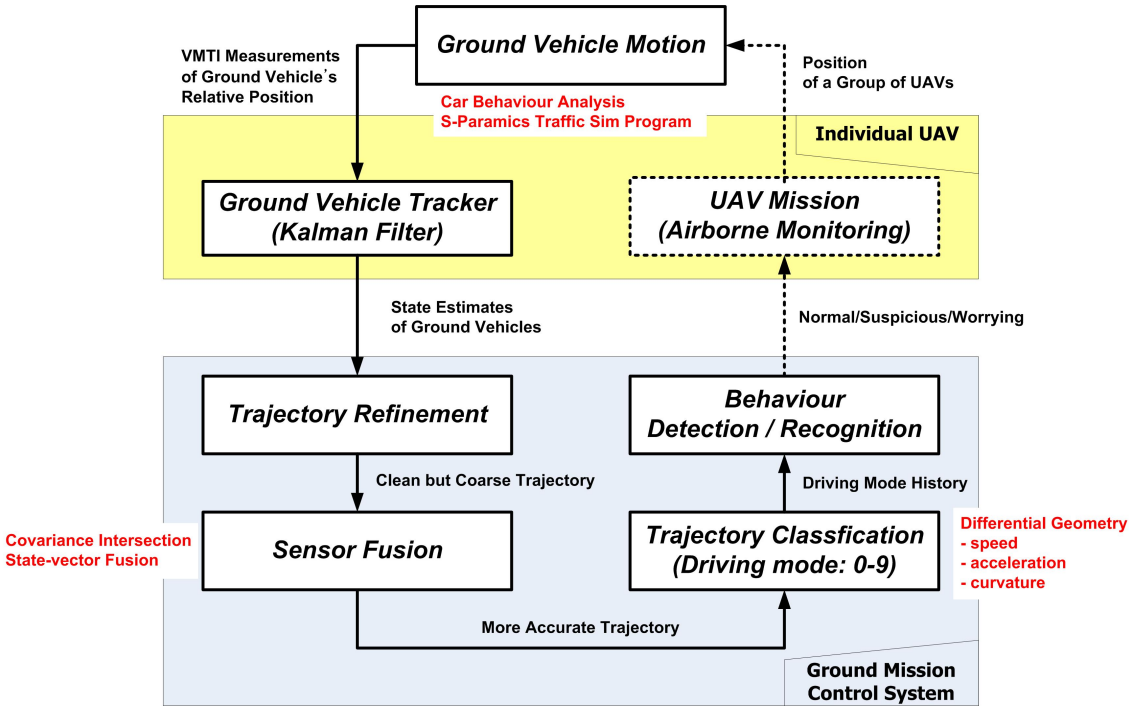


Figure 5.2: Flow chart of overall algorithm for automatic behaviour recognition

classification is followed to model the behaviour of ground vehicles, and lastly behaviour recognition algorithm using string matching theory is presented.

5.2.1 Ground Target Tracking

Assuming a pair of UAVs track the same targets, sensor fusion technique using a Covariance Intersection (CI) algorithm is applied in addition to the Kalman filter. As behaviour description for the ground vehicle requires its trajectory history for a certain length of time, the past state estimates can be used to enhance the tracking accuracy periodically. In this study, an optimal fixed-interval smoother is applied with the Kalman filter. Generally the optimal smoothing algorithm is composed of a forward filter and a backward filter. The basic idea of the smoother is that if the measurements between t and $T_f (> t)$ are available, the estimates of the forward filter at time t, $\hat{x}_f(t)$, can be adjusted based on the estimates of the backward filter at that time, $\hat{x}_b(t)$. Details for the ground target tracking can be found in Chapter 4.

5.2.2 Trajectory Classification

In order to recognise the vehicle's behaviour, the trajectory is classified into driving modes. The purpose of the classification is to categorise the characteristics of manoeuvres associated with forward or lateral driving, by assigning them to driving modes. This classification approach will enable recognition of ground traffic behaviour in a computationally-efficient and flexible way using differential geometric quantities.

In the literature, Fraile and Maybank [121] divided the driving mode of a ground vehicle into four categories: Ahead (forward driving), Left Turn, Right Turn and Stopping. Oliver and Pentland [125] categorised the driving modes into eight manoeuvres: Passing (forward driving), Turning Right, Turning Left, Changing Lanes Right, Changing Lanes Left, Starting and Stopping. Fraile and Maybank's classification is not suitable for the current research because it focuses on low speed car manoeuvres in a urban parking space. On the other hand, Oliver and Pentland's work is more detailed, but they used the sensor signals of a Smart car to classify the driving mode, e.g. analogue speed/acceleration/steering sensors, and digital break/gear indicators. These sensor signals are unavailable to the UAV performing aerial reconnaissance. However, these categories were not suitable to depict the forward (ahead) driving manoeuvres for this application.

Therefore, a new driving-mode classification is proposed, based on references [121, 125] but giving a more detailed approach composed of ten driving modes using speed, curvature, and forward acceleration of the ground vehicle. The assumption for this work is that the driving behaviour persists over a finite length of time rather than from sample to sample time. This implies that the vehicle will perform particular manoeuvre behaviours over a period of time that can be identified. For this, a moving-window-based trajectory approximation [126] is applied using a third-order polynomial function which generates a trajectory with a virtually increased sampling time over a certain time interval. Note that, running a filtering algorithm with an increased sampling time could also be used, providing similar or possibly more accurate trajectory than that of a polynomial approximation. However, this approximation technique can be useful in case that the filter and smoothing algorithms are not working properly due to large process and sensor noises, or unavailable (e.g. only trajectory information with discrete

time step is available without target and sensor model).

Let us assume a new time sequence within a moving window, $0 < T_n < 2T_n < \ldots < (N_T - 1)cT_n = (N_T - 1)T_s$ where T_s is an original sampling time of the tracking filter, T_n is the new virtual sampling time, and N_T is the number of samplings for a moving window. In this study, it is assumed that $N_T = 4$, $T_s = 0.5$, and $c = 5$, and thus the new virtual sampling time is 0.1 seconds. The selection of $N_T = 4$, i.e. a 1.5 seconds moving window, reflects the bandwidth for lane changing of at least 1.0 Hz as described in reference [111] and as shown in Table 5.1.

manoeuvre condition	normal	intoxicated	drowsy	emergency
Time to lane crossing bandwidth	2.5Hz	2.5Hz	1.0Hz	3.0Hz

Table 5.1: Time to lane crossing bandwidth in different manoeuvring situations from [111]

The velocity $(\dot{x}^t(i), \dot{y}^t(i))$ and acceleration $(\ddot{x}^t(i), \ddot{y}^t(i))$ histories with a new time sequence are then used to compute the minimum speed U, the rate of change of orientation $\theta(i)$, and the forward acceleration $a_f(i)$ of the vehicle at the current time step k for each i in a moving window (i.e. $k - c(N-1) + 1 \leq i \leq k$) as:

$$U = \min v(i) = \min \sqrt{\dot{x}^t(i)^2 + \dot{y}^t(i)^2} \tag{5.1}$$

$$\begin{aligned} \theta(i) &= v(i)\kappa(i) \\ &= \sqrt{\dot{x}^t(i)^2 + \dot{y}^t(i)^2}\frac{\dot{x}^t(i)\ddot{y}^t(i) - \dot{y}^t(i)\ddot{x}^t(i)}{(\dot{x}^t(i)^2 + \dot{y}^t(i)^2)^{3/2}} \end{aligned} \tag{5.2}$$

$$a_f(i) = \dot{x}^t(i)\cos\psi(i) + \dot{y}^t(i)\sin\psi(i) \tag{5.3}$$

where κ is the curvature, and $\psi = \tan^{-1}(\dot{x}^t/\dot{y}^t)$ is the heading angle from North. Using above equations, a selective driving mode m_k^d among the driving mode set $M^d = \{0, \cdots, 9\}$ at time step k can be obtained for each moving window with a frequency of $1/T_s$ as:

- **Stopping (0)**, $U < 2$: Since 2 m/s equals to 7.2 km/h, it can be assumed that the car does not move or is about to stop or start moving.
- **Left turn (1)**, $\max(\theta)\min(\theta) > 0$ and $\max(\theta) > \theta_{th,1}$: The inspection of the sign change of θ is used to distinguish a pure turning maneuver from a lane change.

- **Right turn (8)** $\max(\theta)\min(\theta) > 0$ and $\max(\theta) < -\theta_{th,1}$
- **Left lane change (2)** $\max(\theta)\min(\theta) < 0$, $\max(\theta|) > \theta_{th,1}$, and $\theta(0) > 0$: The difference to a left turn is obtained by detecting a sign change of the rate of change of orientation. As one can see in Figure 5.3, the sign of the curvature transits from positive to negative for a left lane change.

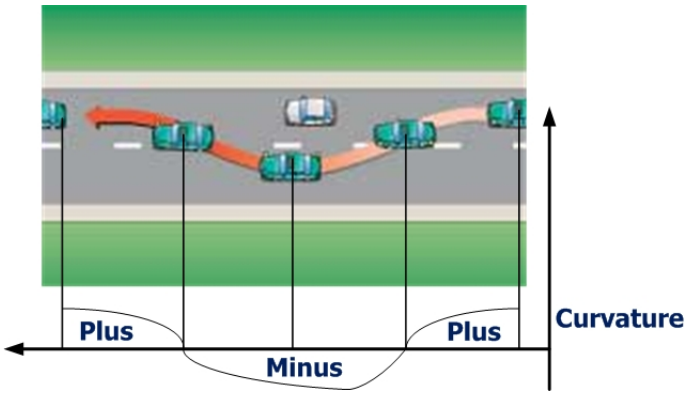


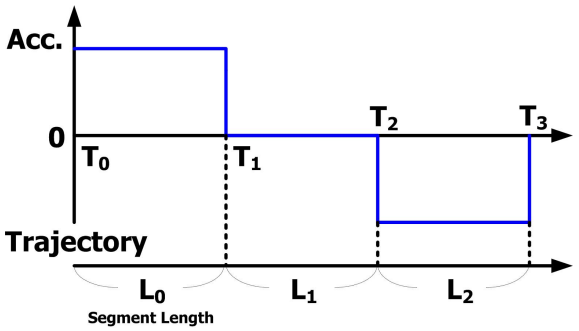
Figure 5.3: The sign of curvature for lane change

- **Right lane change (7)** $\max(\theta)\min(\theta) < 0$, $\max(|\theta|) > \theta_{th,1}$, and $\theta(0) < 0$: As can be seen in Figure 5.3, the sign of the curvature transits from negative to positive in case of a right lane change.
- **U-turning (9)** $\max(|\theta|) > \theta_{th,2}$: If the maximum rate of orientation change is bigger than a certain value, it can be regarded as U-turn manoeuvre performing 180 degree rotation to reverse the direction of travel.
- **Closing gap (6)** $\max(a_f)\min(a_f) < 0$, and $a_f(0) > 0$: Consider a linear trajectory that is sampled at several instances with a known sample period (not necessarily constant) as shown in Fig. 5.4. This illustrates the forward acceleration profile obtained by velocity change over each segment with a length of L. Hence, velocity and acceleration at i-th instance are:

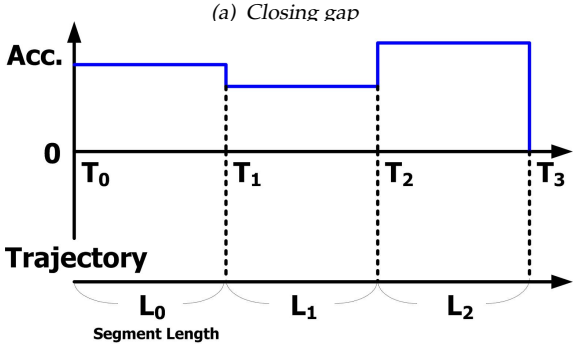
$$v_i = \frac{L_i}{T_{i+1} - T_i} \tag{5.4}$$

$$a_i = \frac{v_{i+1} - v_i}{T_{i+1} - T_i} \tag{5.5}$$

When the driver wants to close gap with the preceding vehicle, the sign of acceleration transits from positive to negative, as shown in Fig. 5.4(a).



(a) Closing gap



(b) Accelerating ahead

Figure 5.4: Acceleration history

- **Widening gap (3)** $\max(a_f)\min(a_f) < 0$, and $a_f(0) \leq 0$: Contrary to the case of gap closing, the sign of acceleration transits from negative to positive for this case.
- **Accelerating ahead (5)** $\max(a_f)\min(a_f) > 0$, and $a_f(0) > 0$: The sign of acceleration stays positive, as shown in the Fig. 5.4(b).
- **Decelerating ahead (4)** $\max(a_f)\min(a_f) > 0$, and $a_f(0) \leq 0$: The sign of acceleration stays negative.

Figure 5.5 shows a full flowchart of the car trajectory classification algorithm proposed in this study.

5.2.3 String Pattern Matching

The essence of the classification is to categorise characteristic manoeuvres associated with forward or lateral driving by assigning them to one of ten classes: $\{0,\ldots,9\}$, as described in the previous section. The next step is to recognise behaviour categories that give rise to concern about the intention of the vehicle

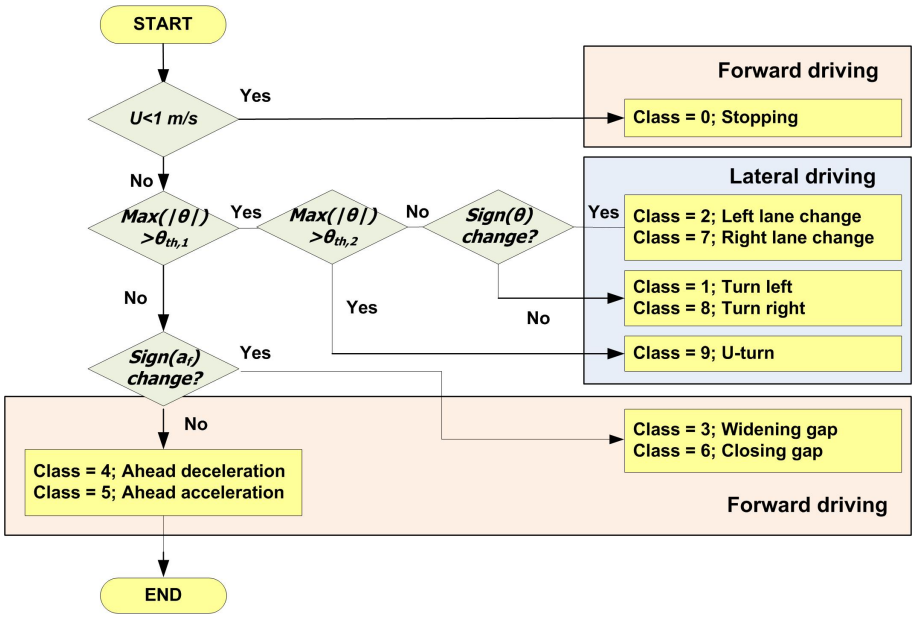


Figure 5.5: Flow chart of driving mode classification

occupants and this study draws on the experience of airborne law enforcement. There exists a large body of knowledge about traffic psychology and behaviour [127, 128] which can be used to study driving behaviour. Within normal traffic, the challenge is to recognize behaviours of potentially dangerous vehicles, disguised as legitimate traffic. Most of these activities are characterized by occasional deviations from characteristics of legitimate traffic. Here the primary problem is how to distinguish these manoeuvres from aggressive or defensive driving patterns of ordinary drivers. Considering these difficulties, a baseline scenario of this study initially deals with the detection and identification of frequently occurring irregular driving behaviour. For this to be achieved, the system should be able to provide continuous tracking of the vehicles of interest and thus enable positive identification of suspect vehicles. By using the simplification of the target characteristic using the driving mode classification, we can apply a string pattern matching algorithm to identify the irregular or suspicious driving behaviour.

The key tools for our behaviour detection scheme are symbolic dynamics and string matching. The mathematical subject of symbolic dynamics originally arose in the theory of dynamical systems and was motivated by the qualitative approach to dynamics in which the character of trajectories is more important than their numerical values. String matching theory is a well-developed area of text

processing. String matching consists in finding all of the occurrences of a string (called a pattern) in a text where the pattern is a string x of length m, while the text is a string y of length n. In this study, using the driving mode set $M^d = \{0, \cdots, 9\}$, a symbolic time series of driving modes $y_k^d = \{m_l^d \in M^d | l = 1, \ldots, N_{sm}\}$ is generated by trajectory classification for each time step k, where N_{sm} represents a moving window length for string matching. The suspicious behaviour is also expressed as strings x_s consisting of ten numbers.

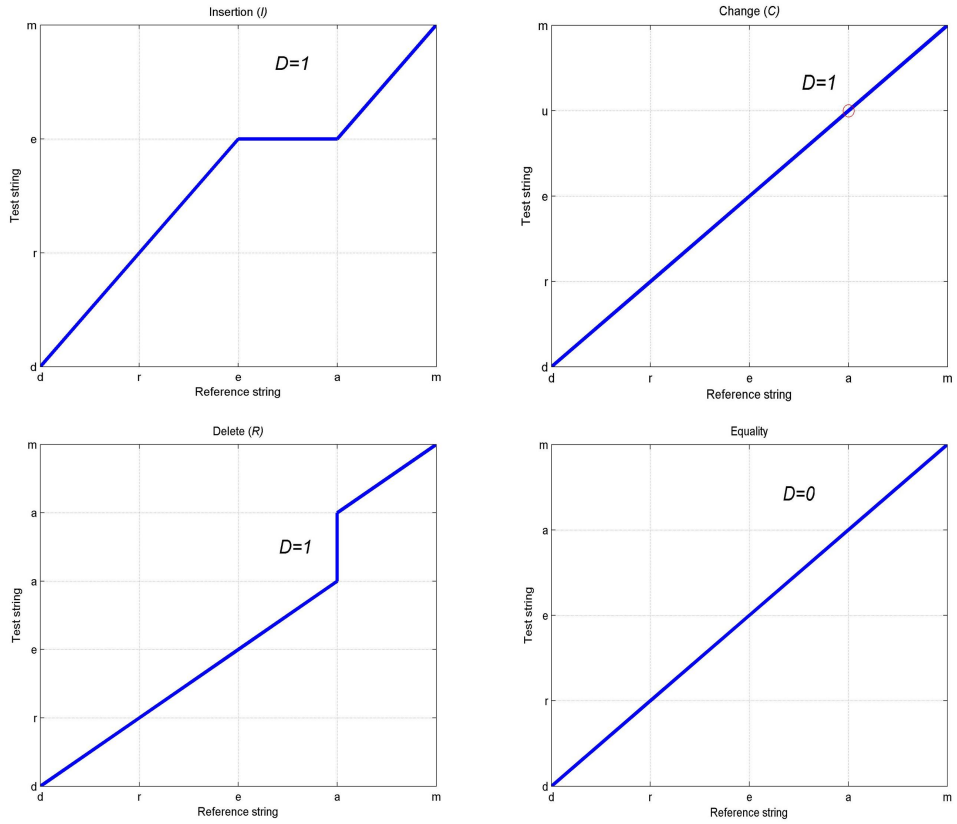
The intuitive string matching method we can apply is an exact matching which detects exactly the same pattern in the driving mode history as the pre-defined suspicious string. However, to find the exact string matching to a certain behaviour class is not technically easy and computationally burdensome. Let us assume that we are interested in a reference pattern (driving mode sequence) '154058', which represents a ground vehicle which turns left, accelerates ahead, decelerates ahead, stops, accelerates ahead again, and lastly turns right. Sometimes, the test pattern appearing as the driving mode sequences, '154458' or '154558' cannot be ignored in the detection scheme, whose fourth element of the string could be one of the following forward driving modes: '3', '4', '5', '6', but not '0'. However, it is not efficient to run the string pattern matching repeatedly using all of these possible driving-mode sequences. Typically this problem arises in speech recognition. When someone says 'beauty', the test pattern may be sensed as 'beety' or 'beaut' due to errors in the reading sensor or of speaking speed difference of the speaker [129]. In this case, we need to define a cost, measuring the distance of the similarity between the reference pattern and the test patterns, which is here defined as the Edit Distance.

The Edit Distance between two patterns is defined as the cost of converting one pattern to the other. If the patterns are of the same length, then the cost is directly related to the number of symbols that have to be changed in one of them to obtain the other pattern. In case the two patterns are not of equal length, symbols have to be either deleted or inserted at certain places of the test string [129]. Although this problem arises in automatic editing and text retrieval applications, it is worth considering for the detection of driving mode sequences which are similar but not exactly matching the predefined sequences. The Edit Distance [129, 130] between two string patterns S_1 and S_2 is defined as the minimum total number of changes

C, insertions I, and deletions R required to change pattern S_1 into S_2:

$$D(S_1, S_2) = \min_j [C(j) + I(j) + R(j)] \tag{5.6}$$

where j represents all possible combinations of symbol variations in order to obtain S_2 from S_1. Figure 5.6 illustrates how to place the patterns on the grid and examples of the Edit Distance calculation using the word 'dream'.



$$D(S_1, S_2) = \min_j [C(j) + I(j) + R(j)]$$

Figure 5.6: Computation of the Edit Distance with an insertion, a change, a deletion, and an equality [129]

A dynamic programming methodology is employed to compute the required minimum editions. For this, let us form a grid by placing the symbols of the reference pattern along the horizontal axis and the test pattern along the vertical axis, as shown in Fig. 5.7. For optimal path searching using dynamic programming, consider the following constraints [129].

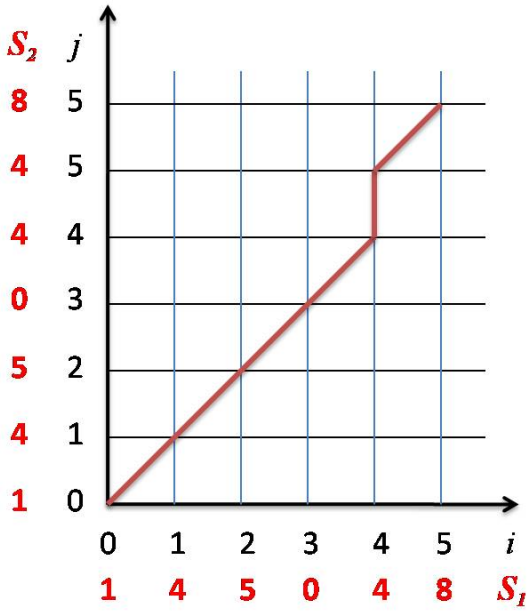


Figure 5.7: Two dimensional grid example for computing Edit Distance

- D(0,0)=0: The cost $D(0,0)$ is zero.
- A complete path is searched.
- Each node (i, j) can be reached only through three allowable predecessors: $(i-1, j), (i-1, j-1), (i, j-1)$.

The cost of the three transitions can be defined for diagonal transitions and horizontal/vertical transitions, respectively.

$$d(i, j|i-1, j-1) = \begin{cases} 0 \text{ if } r(i) = t(j) \\ 1 \text{ if } r(i) \neq t(j) \end{cases} \tag{5.7}$$

$$d(i, j|i-1, j) = d(i, j|i, j-1) = 1 \tag{5.8}$$

Using these constraints and the Edit Distance as the performance index, dynamic programming can be applied to identify the minimum Edit Distance. The proposed detection scheme has a good theoretical basis on symbolic dynamics, and it is intuitive, robust within threshold bounds, computationally efficient and flexible since string patterns to search for can be easily adapted.

Note that, detecting a cyclical/circular shift of a reference string could be also of great importance for a suspicious string consisting of a set of repeated

driving modes. By comparing a pre-defined suspicious string as well as its circularly shifted strings with the driving mode history of a certain length, the proposed behaviour detection scheme could cope with this cyclical/circular shift. For instance, let us assume a driving mode history $a = [4\,5\,6\,1\,2\,3]$, and a suspicious string $b = [1\,2\,3\,4\,5\,6]$. By shifting the string b progressively, Edit distance D will be changed as:

$$\begin{aligned}
D(a, circshift(b,0)) &= D(a,[1\,2\,3\,4\,5\,6]) &= 6\\
D(a, circshift(b,1)) &= D(a,[6\,1\,2\,3\,4\,5]) &= 5\\
D(a, circshift(b,2)) &= D(a,[5\,6\,1\,2\,3\,4]) &= 3\\
D(a, circshift(b,3)) &= D(a,[4\,5\,6\,1\,2\,3]) &= 0\\
D(a, circshift(b,4)) &= D(a,[3\,4\,5\,6\,1\,2]) &= 3\\
D(a, circshift(b,5)) &= D(a,[2\,3\,4\,5\,6\,1]) &= 5
\end{aligned}$$

where $circshift(b,i)$ circularly shifts the values in the string b by the shift size i. Examining the minimum Edit distance D amongst all shifted strings makes it possible to detect the suspicious string having aforementioned shifting characteristic in the driving mode history.

5.2.4 Numerical Simulations

For the tracking of ground vehicles using real VMTI measurements, it is assumed that the noise covariances of the x and y position are $(\sigma_x^2, \sigma_y^2) = (1\ m^2, 1\ m^2)$. Figure 5.8 shows the UAV and ground vehicle trajectories obtained from VMTI measurements provided by Roke Manor, overlaid onto a Google earth satellite map [3]. Even though there are much more ground vehicles in the data from Roke Manor than those in the figure, we selected and used some of them which have a consistent target ID and with a reasonably long trace, as shown by the red lines.

Note that '3: WIDENING GAP' and '6: CLOSING GAP' modes are replaced with '3: DEC→ACC' and '6: ACC→DEC', respectively since we can not identify the preceding vehicle in this scenario. These modes would be useful to capture abrupt and frequent braking as is dangerous manoeuvre. Since there is no suspicious or anomalous behaviour in the VMTI data, a behaviour string x_s is set to '6 6 3 3 4 4' (6: ACC→DEC, 3: DEC→ACC and 4: Deceleration ahead) is used to validate the feasibility of the detection algorithm. More suspicious behaviours

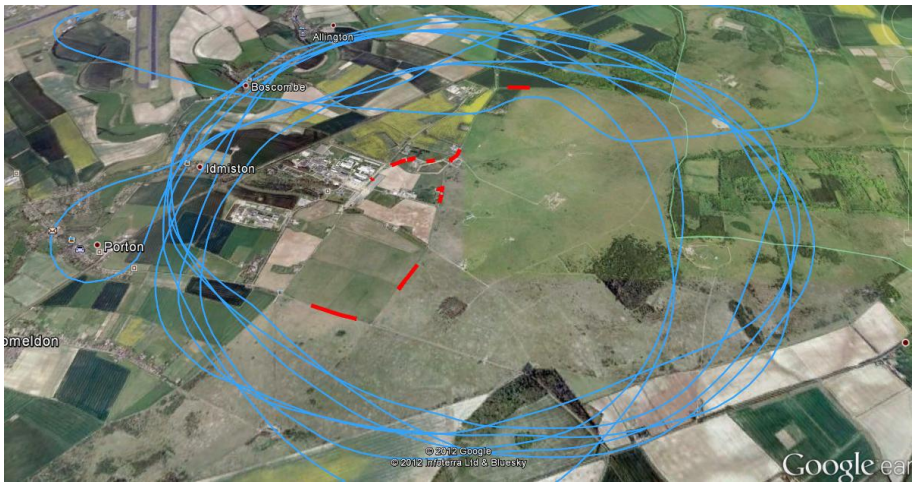


Figure 5.8: UAV (blue line) and ground vehicle trajectories (red lines) from VMTI measurements provided by Roke Manor with GIS satellite data overlaid thanks to Google earth [3]

will be tackled in the next section, including evasive, rendezvous and retreating manoeuvres. A driving mode pattern is regarded as *suspicious* behaviour if it has less than an Edit Distance of four and *worrying* behaviour if it has less than an Edit Distance of three.

Figures 5.9~5.10 show the simulation result using VMTI sensor measurements including trajectory estimation, the trajectory on a Google map, minimum speed and maximum rate of heading change in a moving horizon, trajectory classification and behaviour recognition. Since the VMTI measurement data do not contain significant noise (resulting from the image processing of Roke Manor), estimated trajectory from the Kalman filter looks similar to the measurements themselves, as shown in Fig. 5.9(a). However, the estimation process is still required to obtain smooth and continuous position, velocity and acceleration estimates of the ground vehicle. The classification results combined with the geographical context shown in Figs. 5.9~5.10 are seen to be promising, in the sense that they capture a real-world situation reasonably well. The resulting behaviours are identified: right lane change (or abrupt direction change when the vehicle starts to move) (2), speed up (5), repetition of acceleration and deceleration to control the speed (6), deceleration to prepare turning for the road ahead, and finally turning right (1). Hence the driver's behaviour over 20 seconds is represented by a simple string of numbers '0 2 5 6 1'. The approximate string matching successfully detects

suspicious behaviours similar to the '6 6 3 3 4 4' string, such as '6 6 6 4 4 4' or '6 6 4 4 4 4' at approximately 14 second.

Figure 5.11 shows the results of the trajectory classification and behaviour recognition process for another vehicle from the VMTI sensor measurement data. Even if there are no measurements for about 30~35 seconds, the ground target tracker provides a reasonable estimate of the state and re-acquires the target successfully, with the help of the prediction process in the Kalman filter, as described in Eqs. (4.18)~(4.22). It can be conjectured from the fact that there is no sudden change of speed or curvature right after the measurement drop out. In this case, since the vehicle is moving on a rural road (or farm ground), as shown in Fig. 5.11(b), forward driving modes (3, 4, 5, 6) were used frequently to maintain slow constant speed, compared to vehicles on the road.

Furthermore, as can be seen in Fig. 5.9~5.12(f), it is only necessary for the operator to check a few representative strings in order to identify the suspicious vehicles. This can reduce the required human workload significantly, as well as enhance the efficiency and success rate of detecting suspicious vehicles for the operator monitoring complex traffic flow in the region of interest. For instance, if there are more than a hundred vehicles in the region of interest under surveillance, the behaviour recognition results for all vehicles can be displayed on one monitor screen. If some of vehicle's automated recognition results are either suspicious or worrying, the operator can investigate those more thoroughly, so that situational awareness will be built up in a fast and efficient manner. Combining this approach within a spatiotemporal context, designating specific areas and times of the day as having increased or decreased sensitivity, can be of great help to confirm the final decision for suspicious vehicles. This is because traffic behaviour will significantly change, depending on the condition of surrounding environment, such as the road type or current traffic flow. This will be considered in the following section.

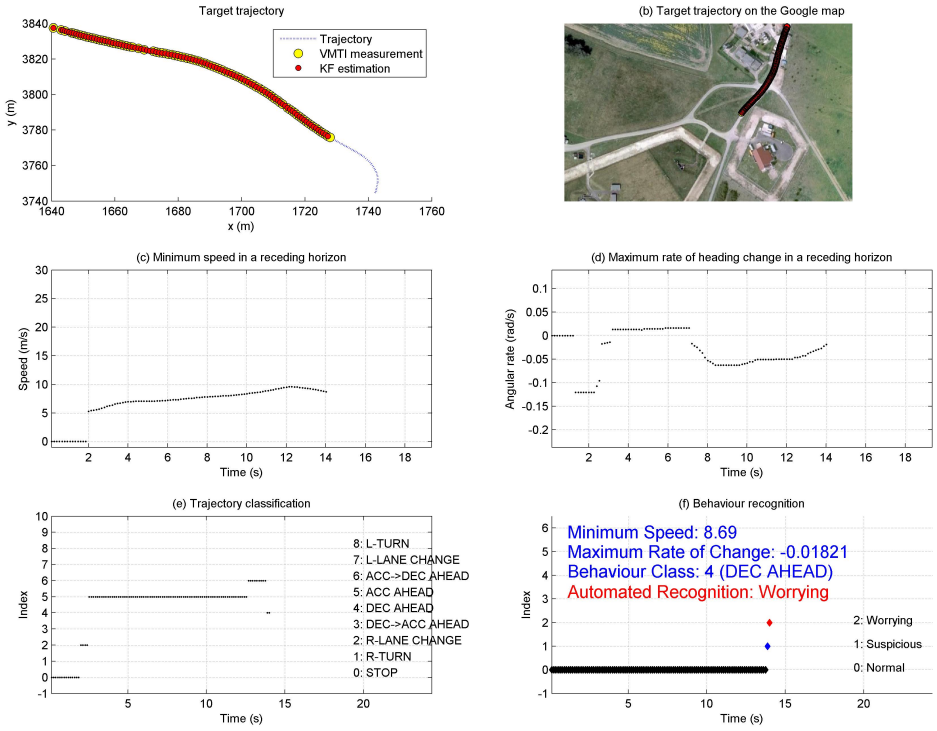


Figure 5.9: Integration results of automatic behaviour recognition with VMTI Information (Target ID=222 at t=14s)

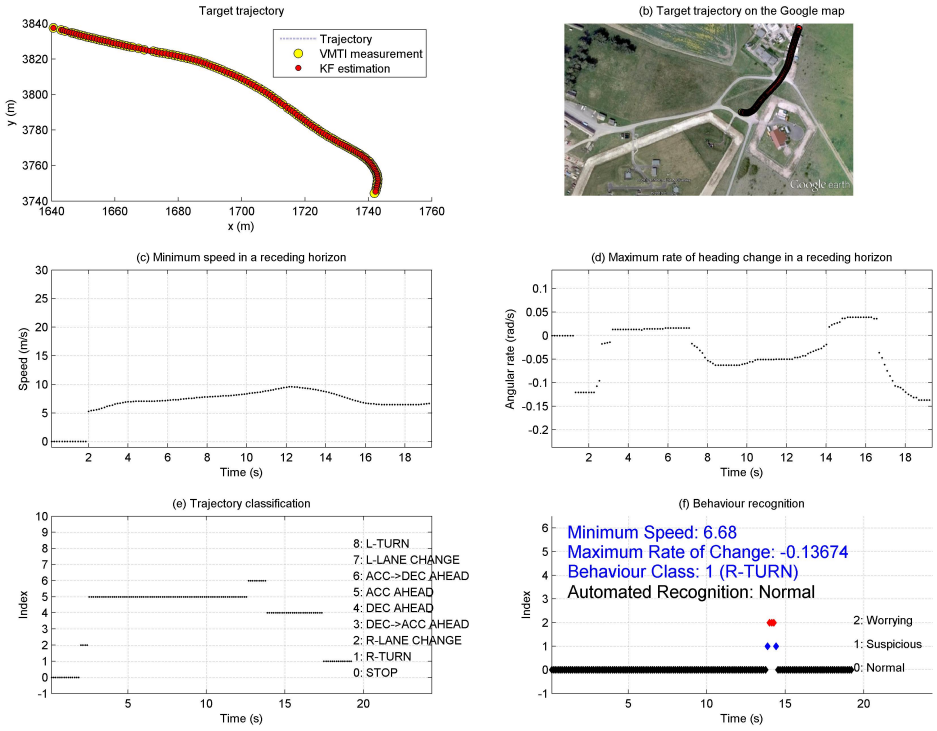


Figure 5.10: Integration results of automatic behaviour recognition with VMTI Information (Target ID=222 at t=19s)

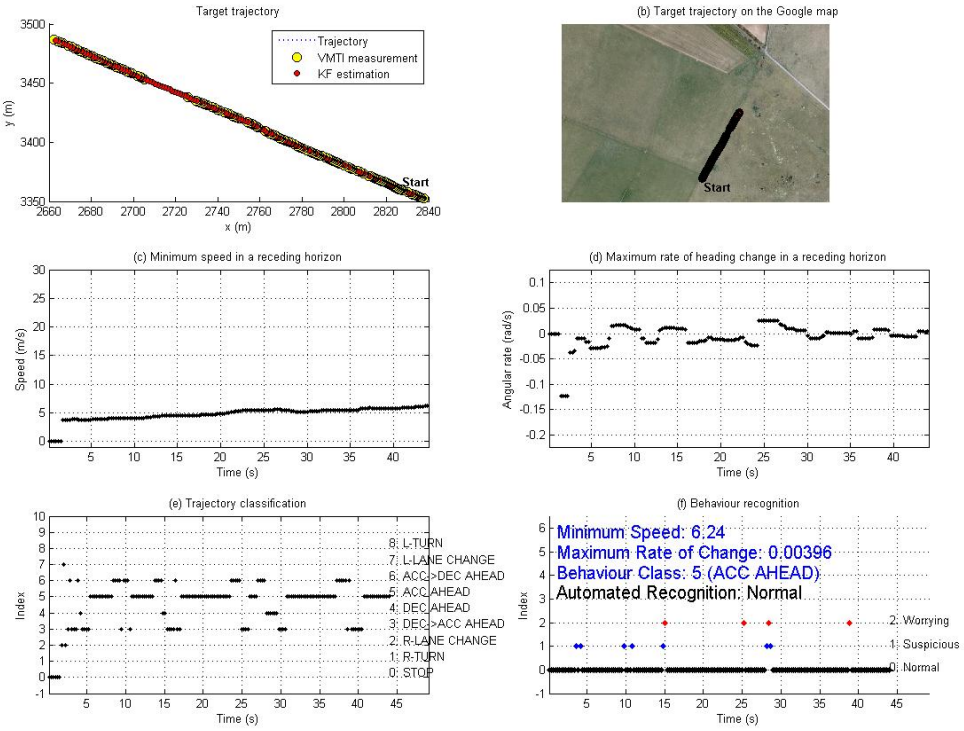


Figure 5.11: Integration results of automatic behaviour recognition with VMTI Information (Target ID=372)

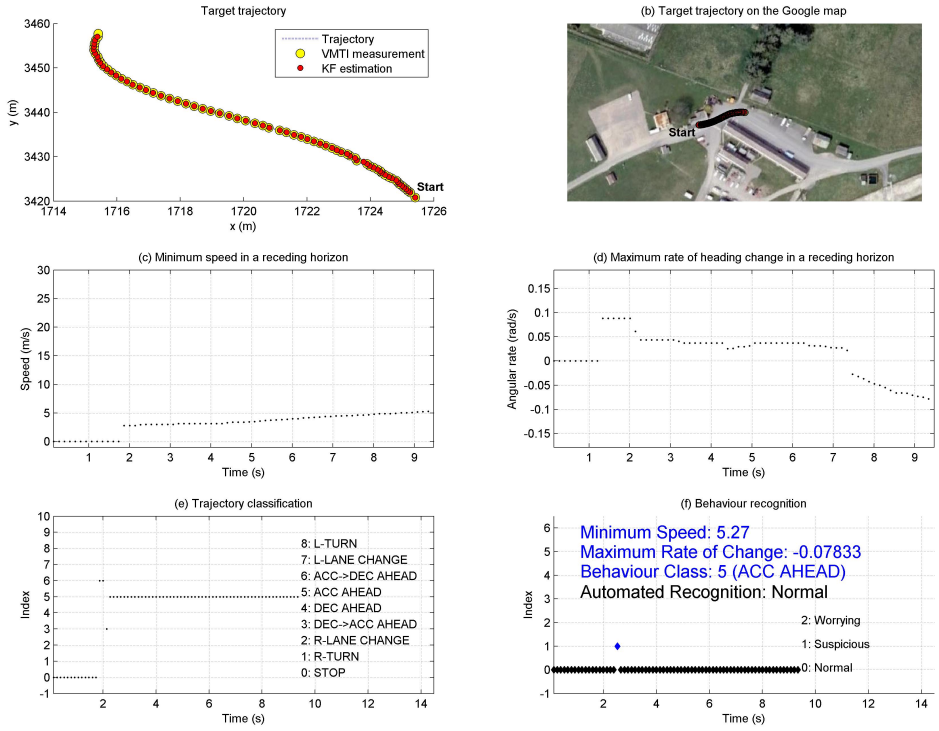


Figure 5.12: Integration results of automatic behaviour recognition with VMTI Information (Target ID=64)

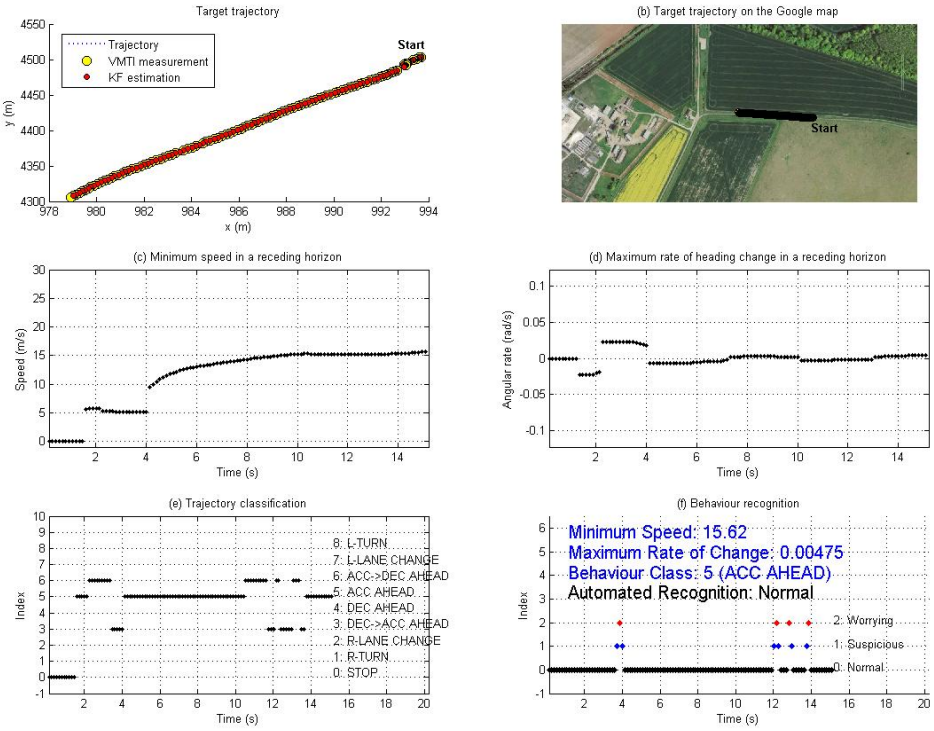


Figure 5.13: Integration results of automatic behaviour recognition with VMTI Information (Target ID=214)

5.3 Fuzzy Rule-Based Decision Making

This section describes a fuzzy expert rule-based airborne monitoring methodology to detect ground vehicle behaviour as an extension to our previous algorithms. In previous sections, the primary source for the behaviour recognition is a single deterministic behaviour cost, obtained from an approximate string matching. Although this cost (defined as Edit Distance) can provide a measure of detection by computing the similarity between pre-defined suspicious strings and driving mode history, additional information needs to be considered to finally confirm the characteristic of behaviour while avoiding frequent false alarms. Therefore, in this study, to systematically exploit all available information obtained from complex environments, a fuzzy system is applied because of its ability to classify complex sources into simple and intuitive forms in the form of a rule-base. The proposed fuzzy expert rule-based decision making allows concurrent accommodate of several aspects of behaviour as well as spatiotemporal environment factors, aided by supervision of a human operator. An overall flow chart of the technique presented in this study is shown in Fig. 5.14. Note that this figure represents the same structure as Fig. 5.2 but with additional environmental aspects and a fuzzy decision making block.

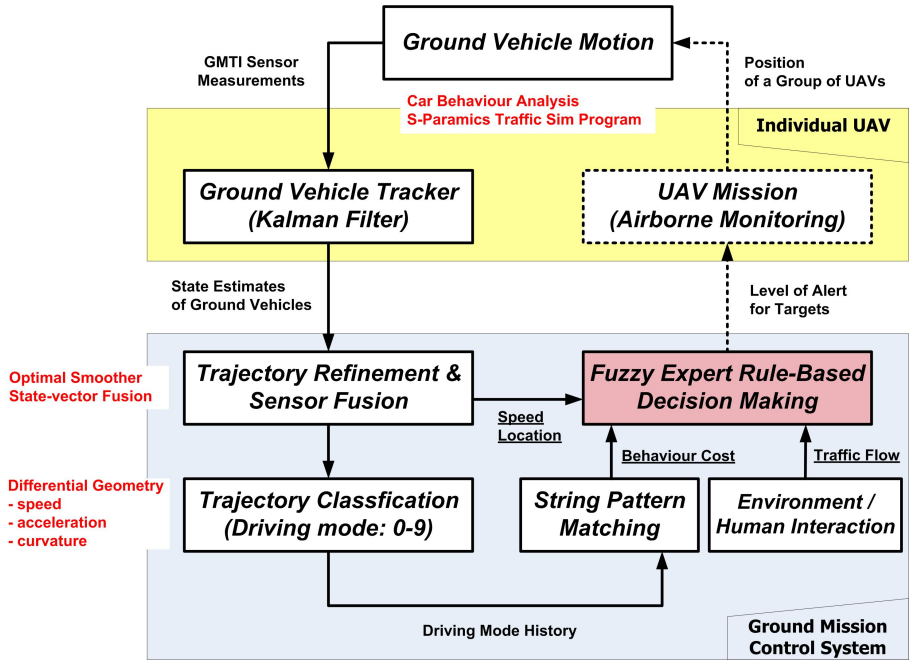


Figure 5.14: An overall flow chart of fuzzy expert rule-based behaviour monitoring

To systematically exploit all available information, a fuzzy system first classifies all available input information using membership functions (fuzzification), applies a rule-base and then produces an output result based on the rules (defuzzification). The fuzzy system used in this study consists of four fuzzy membership functions to classify the inputs and one output constructed from 36 expert inference rules, as shown in Fig. 5.15. Note that, the values in membership functions and inference rules used in the following are for illustration only, rather than extracted from experiments or experts.

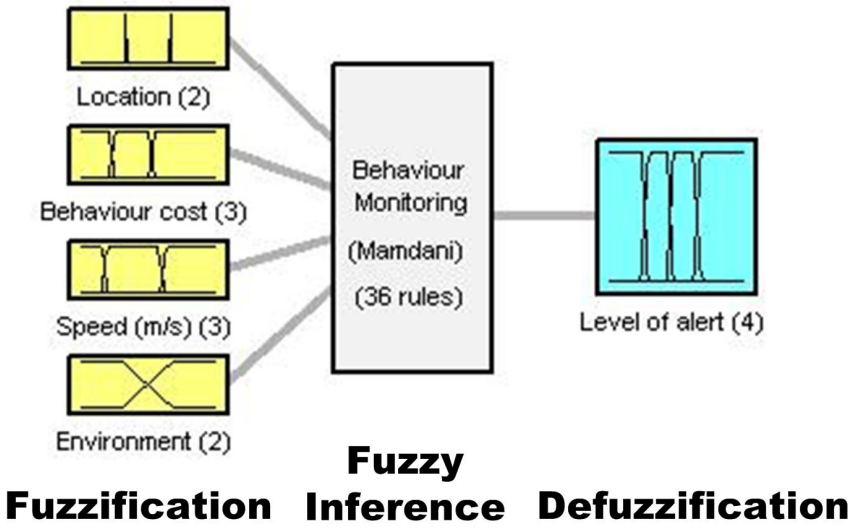


Figure 5.15: Structure of fuzzy decision making

5.3.1 Fuzzification

The fuzzy input for behaviour monitoring includes four parameters: location, behaviour cost, speed of the vehicle, and environmental data composed of:

- **Location:** A time history of the position of the suspicious ground vehicle relative to a critical area (e.g. the centre of complex activities or the base walls of military facilities) or an index of the roadmap that the ground vehicle has moved along. Assuming that the local roadmap information is readily available in advance, the indexes of the local roads in the region of interest can be annotated by a sequence of road numbers. If the vehicle

travelling on one of identified roads of interest, the location is categorised as 'Region of interest (R)'; otherwise it is categorised as 'General (G)', as shown in Fig. 5.16(a).

- **Behaviour cost:** As a key factor in behaviour monitoring, the Edit Distance D from specified test strings can be used to provide a time history of the behaviour cost. Let $X_s = \{x_s^1, \cdots, x_s^{N_{su}}\}$ be the set of pre-defined suspicious behaviours. Then, the behaviour cost C_k^s with respect to current time series of driving modes y_k^d and suspicious behaviours at time step k can be defined as:

$$C_k^s = \frac{1}{\min_{p \in X_s} D\left(x_s^p, y_k^d\right) + 1} \tag{5.9}$$

A multi-vehicle interaction event, such as a vehicle stopping roughly in the same place at the same time, needs to be considered in the Behaviour Cost. This sort of rendezvous event is detected by checking the distances between nearby ground vehicles which are stopped. For the i^{th} vehicle, the distance is first calculated with all other vehicles in $GV_k^{stop} = \{gv_k^1, gv_k^2, \cdots, gv_k^{n_{st}}\}$ which is a stopped vehicle set at time step k.

$$d_k^i = \min \sqrt{(x_k^j - x_k^i)^2 + (y_k^j - y_k^i)^2} \quad \forall\ j \in GV_k^{stop}. \tag{5.10}$$

Then, the behaviour cost of the i^{th} vehicle for rendezvous is obtained using distance d_k^i:

$$C_k^{i,r} = e^{-\beta d_k^i} \tag{5.11}$$

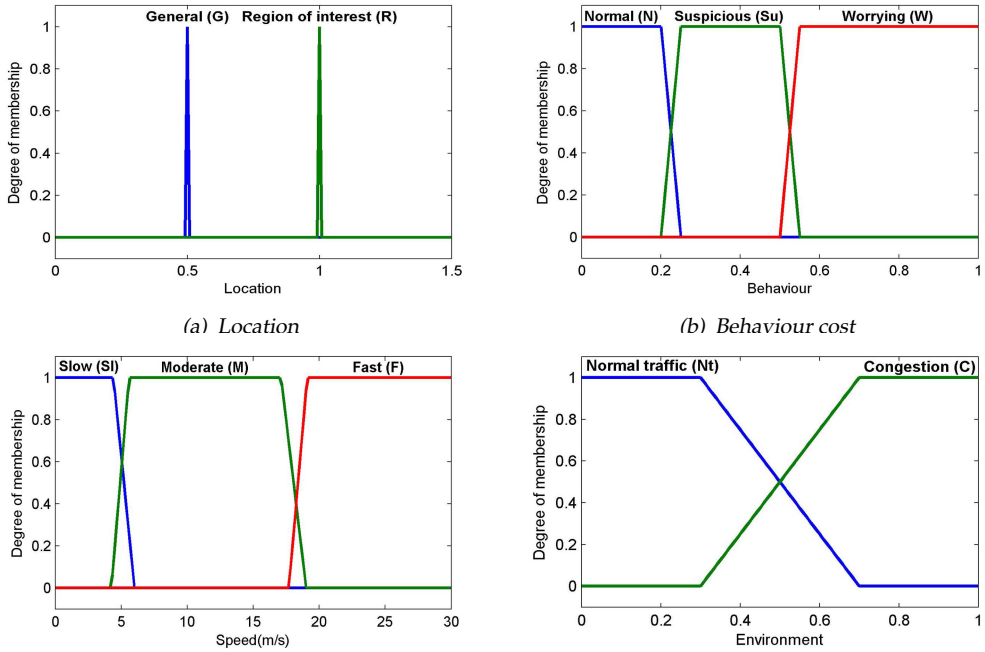
where β is a positive gain parameter. The final Behaviour Cost $C_k^{i,b}$ of i^{th} vehicle is then calculated as:

$$C_k^{i,b} = \max\{C_k^{i,s}, C_k^{i,r}\} \tag{5.12}$$

Three fuzzy membership functions defined with linguistic variables 'Normal (N)', 'Suspicious (Su)', and 'Worrying (W)' are used to categorise the Behaviour Cost, as shown in Fig. 5.16(b). Note that this Behaviour Cost can be replaced by other costs such as the anomaly score from the Gaussian Process as explained in the previous section instead of Edit Distance.

- **Speed:** The speed of the vehicle with respect to its position or time also needs to be investigated since it can provide a measure of suspicious or abnormal behaviour. A membership function with linguistic variables 'Slow (Sl)', 'Moderate (M)', and 'Fast (F)' are used, as shown in Fig. 5.16(c).

- **Environment:** The last input considers an environmental condition with human interaction for the behaviour decision process. Depending on the traffic flow density, two membership functions with linguistic variables 'Normal traffic (Nt)' and 'Congestion (C)' are used, as shown in Fig. 5.16(d). Even though only traffic flow is used in this study, it can be easily replaced with time zone (day/night or weekday/weekend), or any other environmental parameters. Human operator input can also be used in the rule-base instead of relying on an autonomous decision process which can be vulnerable to unexpected and dynamic environments.



(a) Location *(b) Behaviour cost*

(c) Speed *(d) Environment (Traffic flow)*

Figure 5.16: Membership functions for fuzzy inputs

The fuzzy output for behaviour monitoring is the level of alert for each ground vehicle, consisting of a membership function with four linguistic variables, 'Allow', 'Monitor', 'Investigate' and 'Respond', as shown in Fig. 5.17.

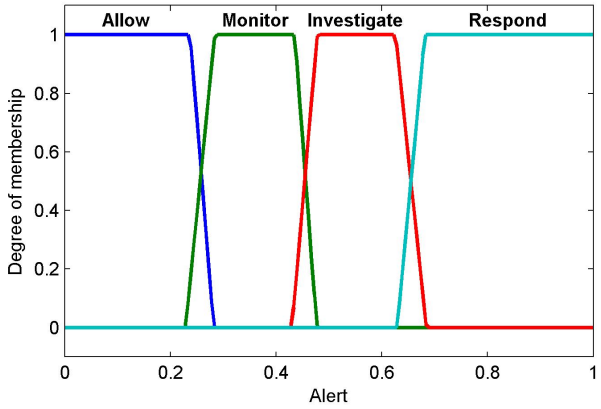


Figure 5.17: Membership function for fuzzy output

5.3.2 Fuzzy Inference

The fuzzy inference system is designed using a Mamdani model [131]. Expert knowledge can be expressed in a natural way using the linguistic variables previously defined to create a rule-base shown in Table. 5.2~5.3. From the table, the rules can be interpreted as:

- Rule 1: If Location is 'G' and Behaviour is 'N' and Speed is 'Sl' and Environment is 'Nt', then Alert is 'Allow'.

Table 5.2: Fuzzy rule 1~18: location is 'G' (General road)

Rule No.	*Behaviour*	*Speed*	*Environment*	*Alert*
1 / 2:	N	Sl	Nt / C	Allow / Allow
3 / 4:	N	M	Nt / C	Allow / Allow
5 / 6:	N	F	Nt / C	Monitor / Investigate
7 / 8:	Su	Sl	Nt / C	Monitor / Allow
9 / 10:	Su	M	Nt / C	Monitor / Allow
11 / 12:	Su	F	Nt / C	Investigate / Investigate
13 / 14:	W	Sl	Nt / C	Investigate / Monitor
15 / 16:	W	M	Nt / C	Investigate / Monitor
17 / 18:	W	F	Nt / C	Respond / Respond

Table 5.3: Fuzzy rule 19~36: location is 'R' (Region of interest)

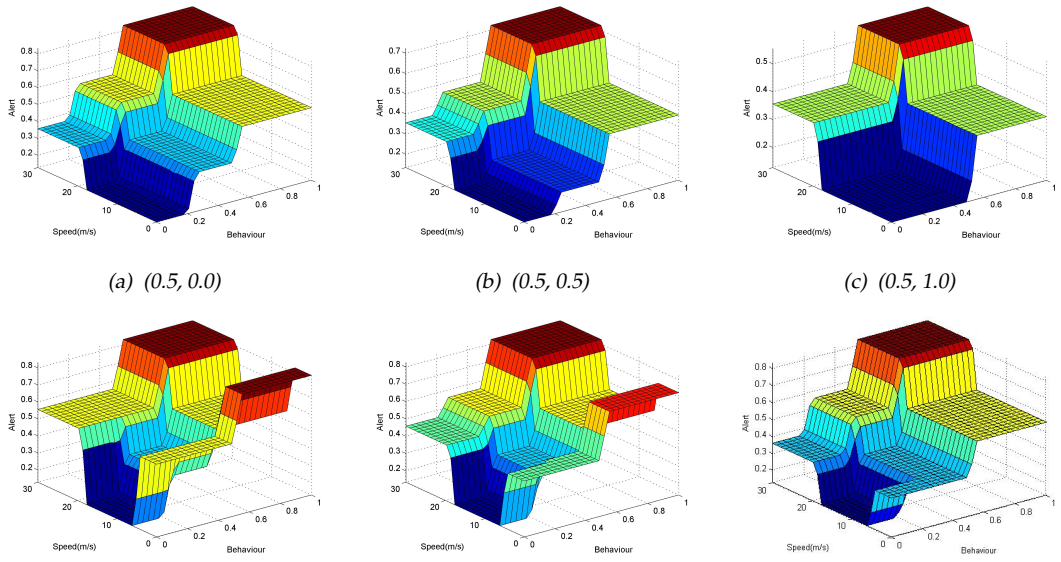
Rule No.	*Behaviour*	*Speed*	*Environment*	*Alert*
19 / 20:	N	Sl	Nt / C	Investigate / Monitor
21 / 22:	N	M	Nt / C	Allow / Allow
23 / 24:	N	F	Nt / C	Investigate / Monitor
25 / 26:	Su	Sl	Nt / C	Investigate / Monitor
27 / 28:	Su	M	Nt / C	Monitor / Monitor
29 / 30:	Su	F	Nt / C	Investigate / Investigate
31 / 32:	W	Sl	Nt / C	Respond / Investigate
33 / 34:	W	M	Nt / C	Investigate / Investigate
35 / 36:	W	F	Nt / C	Respond / Respond

Note that, depending on the location and the environment, the rules are modified. For instance, if the location of the vehicle is 'G' (i.e. general area), the speed 'Sl' does not mean something significant, thus leading to an 'Allow' output, whereas if the location is 'R' (i.e. region of interest), slow speed or stopping of the vehicle can be identified as suspicious (monitoring the military base or placing of improvised explosive devices) leading to an 'Investigate' output, using Rule 1 and 19. However, even though the location is 'R' and the speed is 'Sl', if the environment is 'C' (i.e. congestion), its alert level should be alleviated using Rule 20 since slow speed is more likely to be observed in this area.

5.3.3 Deffuzification

Using the input variables and the defined fuzzy rules, the fuzzy outputs for all rules are then aggregated into one output fuzzy set. Finally, to obtain a crisp decision value for the level of alert, a defuzzification process needs to be performed. Even though there are several algorithms for this defuzzification, this study uses the method of taking the centre of gravity of the aggregated output fuzzy set [132]. Figure 5.18 shows the recommended level of alert using the fuzzy input variables with defined inference rules. From Figs. 5.18(a)~(c), it can be observed that the level of alert tends to decrease as the environment value (i.e. traffic flow density) increases when the location is 'G' (equivalently, 0.5). Meanwhile, in case

the location is 'R' (equivalently, 1.0), as shown in Figs. 5.18(d)~(f), representing the vehicle moving in a region of interest, slow speed causes a high alert, as shown in Fig. 5.18(d). Its effect decreases as traffic flow increases.



(a) (0.5, 0.0) *(b) (0.5, 0.5)* *(c) (0.5, 1.0)*

(d) (1.0, 0.0) *(e) (1.0, 0.5)* *(f) (1.0, 1.0)*

Figure 5.18: Recommended level of alert according to input variables (Location, Environment)

5.3.4 Numerical Simulations

This section presents numerical simulations for both a military and civilian traffic scenario, using the proposed fuzzy expert rule-based monitoring algorithm with two UAVs loitering over a certain area.

5.3.4.1 Military scenario

Figure 5.19 shows the map description where the ground vehicles are moving around a military base. In the map, at the southern area of a river, there is a military base of strategic importance to be protected, which has a surrounding roadmap used by civilian ground vehicles, which has roads near the base wall. In this scenario, only road 3 and 4 are assumed to be of interest (i.e. the location is 'R'), and the suspicious behaviour string x_s^{stop} '4 4 0 0 0 0' is selected (which

is interpreted as deceleration and then stopping) to detect if the vehicle stops around the military base. To detect further potential threats (such as a vehicle placing an IED), another suspicious behaviour string set x_s^{threat} is selected as '9 9 9 9 9 9', '9 9 9 9 9 1' ~ '9 9 1 1 1 1' (9: U-turn and 1: R-turn). This threat string is valid once string x_s^{stop} is detected since a U-turn manoeuvre is likely to occur after stopping for IED placement. In addition, the size of driving mode history y^D is set to $N_{sm} = 6$ which is the same as that of the test sets x_s. Driving mode patterns are regarded as suspicious with an Edit Distance of less than four and worrying with an Edit Distance of less than three, and the corresponding Behaviour Cost is determined using Eq. (5.9) as the fuzzy input.

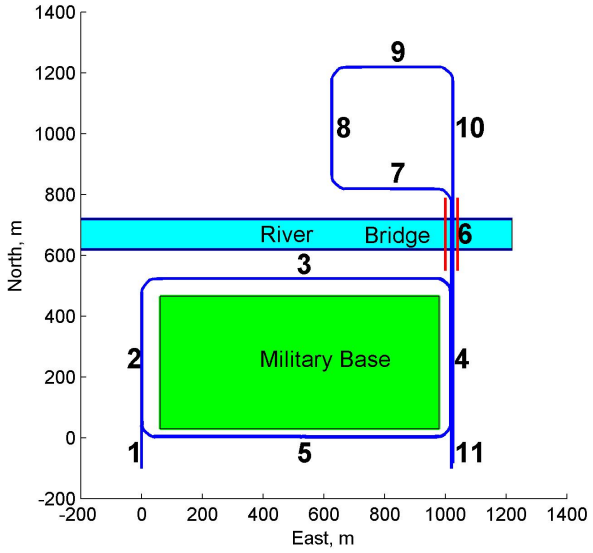


Figure 5.19: Map of military scenario

Figure 5.20 shows result of behaviour recognition of ground target 1 including the trajectory, the time history of the road index, the minimum speed in a receding horizon, the trajectory classification, behaviour recognition, and the fuzzy rule-based decision making. In this case, the ground vehicle first circles clockwise around the military base. During that time, the vehicle stops for 15 seconds in the middle of road 3 at approximately 420s. After that, it reverses its direction of travel (i.e. performing a U-turn) and then travels back to where it came. The trajectory classification histories show a reasonable performance, capturing the turning or stopping manoeuvre in conjunction with road index history, as shown in Fig. 5.20(d). The behaviour recognition algorithm using the Edit Distance suc-

cessfully detects the stopping manoeuvre and the following U-turn manoeuvre, as shown in Fig. 5.20(e). In normal traffic, shown as the blue line in Fig. 5.20(f), the level of alert output from the fuzzy decision making process has a high value when the location is 'R', the Behaviour Cost is high and the speed is slow. However, if there is congestion in the traffic, the effect of the Behaviour Cost and the slow speed on the level of alert is reduced shown as the red line in Fig. 5.20(d), since those input conditions are more likely to happen due to traffic congestion.

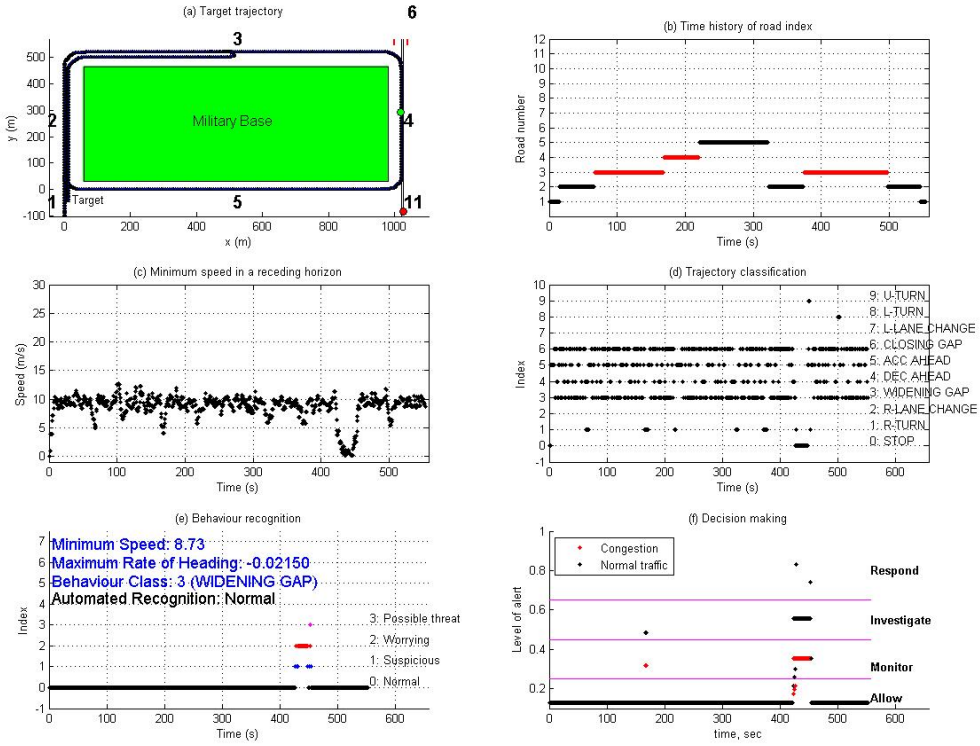


Figure 5.20: Behaviour recognition result of ground target 1 for military scenario: possible threat-placing IED

Figure 5.21 shows the result of behaviour recognition of ground target 2. When the vehicle stops at about 120s and since the vehicle is on road 7 (which is location: 'G'), the level of alert is lower compared to the previous case where the target stopped on the road of interest. Lastly, Fig. 5.22 shows the situation where the vehicle slows down on the road of interest. Even though there is no obvious suspicious behaviour for 500~600s, since the speed of the vehicle is slow on the road 4 (which is location 'R'), the fuzzy logic rule-base gives a high level of alert.

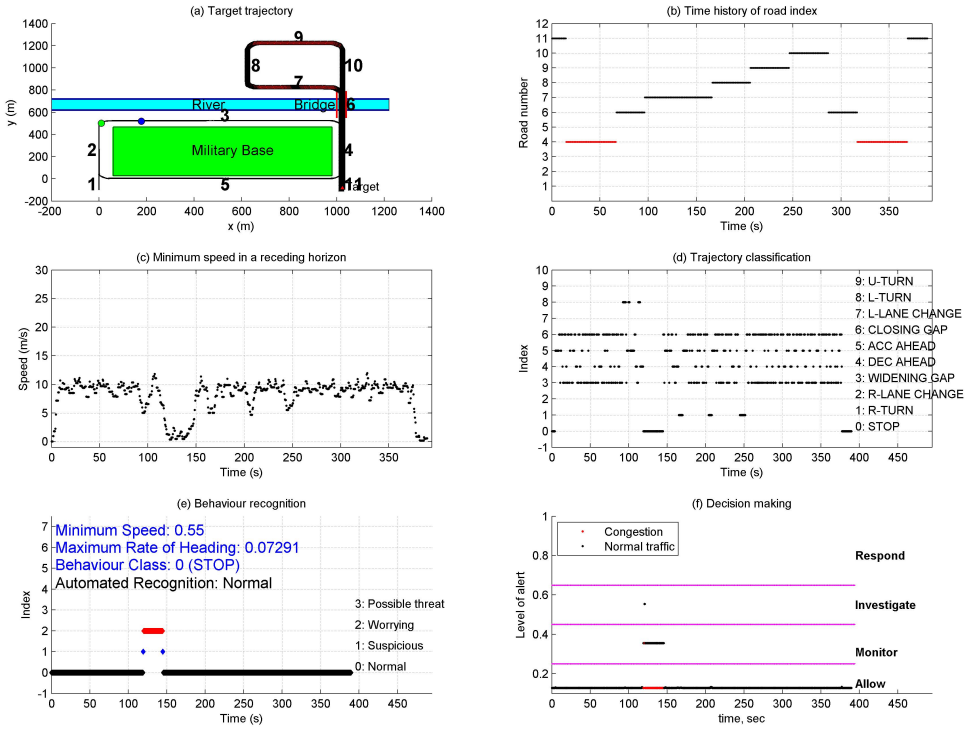


Figure 5.21: Behaviour recognition result of ground target 2 for military scenario: stopping

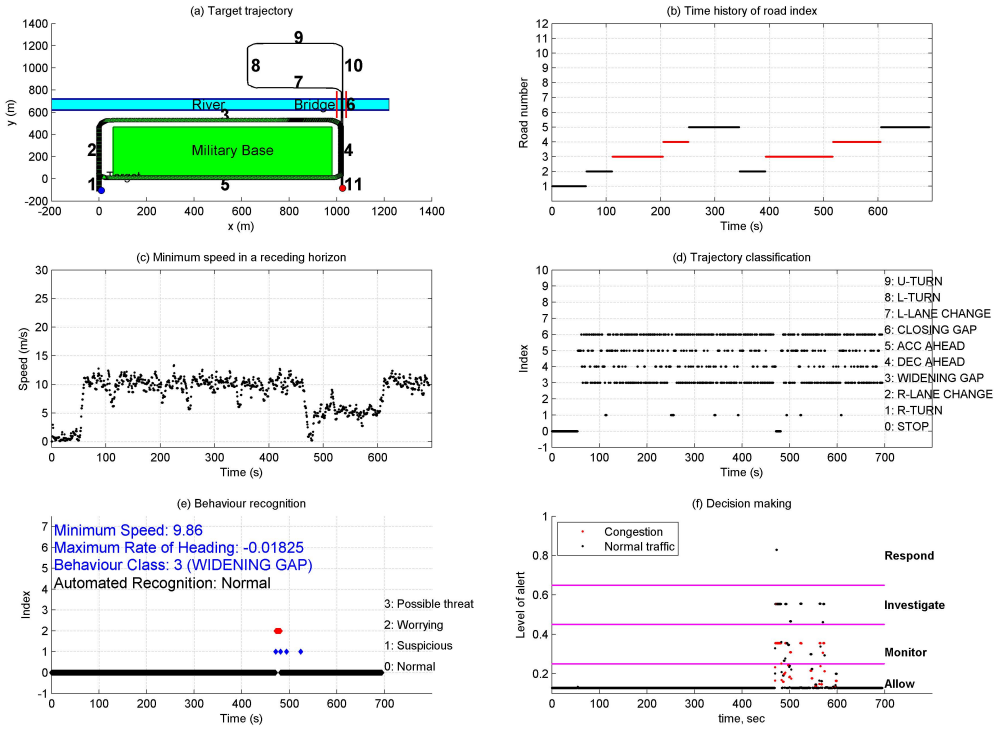


Figure 5.22: Behaviour recognition result of ground target 3 for military scenario: slow speed surveillance

5.3.4.2 Civilian traffic scenario

A large number of ground target trajectories are obtained using an S-Paramics [94] traffic model of Devizes in the UK, sampled at 2 Hz, as shown in Fig. 5.23, and activities inside the white box are selected as being of interest. Using the S-Paramics data, frequent lane changes are inserted artificially for one vehicle, in order to generate suspicious behaviour. This manoeuvre is labelled as *weaving* or *evasive*, and can be viewed as one of the most dangerous behaviours in civilian traffic. To detect this, suspicious behaviour string x_s '2 7 2 7' and '7 2 7 2' are selected (2: right lane change and 7: left lane change). A stopping manoeuvre is also inserted to simulate a multi-vehicle interaction, i.e. a rendezvous event. In this scenario, the driving mode pattern is regarded as suspicious behaviour with an Edit Distance (with respect to the suspicious behaviour x_s) of less than three and worrying behaviour with an Edit Distance of less than two. In addition, every road is assumed to be a general road (i.e. location is 'G').

Figures 5.24~5.26 show the result of behaviour recognition of ground target 1 including the trajectory, the nearby ground traffic, the trajectory classification, behaviour recognition, and the fuzzy rule-based decision making. As can be seen in Fig. 5.24(d), the string pattern matching algorithm detects the evasive manoeuvre successfully over a period of 10~20 seconds. As ground target 1 (red) and ground target 2 (blue) both stop in roughly the same place at the same time, the rendezvous event is also detected with its cost, as shown in Fig. 5.26(d) and Fig. 5.27(d). In normal traffic shown by the blue line in Fig. 5.26(e), the level of alert has a high value when the Behaviour Cost is high, which means that evasive manoeuvre or rendezvous between vehicles is likely. In case of congested traffic as shown by the red line in Fig. 5.26(e), although the level of alert shows the same tendency as the previous example, the effect of the Behaviour Cost is reduced as frequent lane change or stopping is more likely to happen due to traffic congestion. For ground target 3, even though the rendezvous event occurs with ground target 1 at approximately 40s, since the Behaviour Cost is small due to the large distance between them, this event results in very low level of alert, as shown in Fig. 5.28. By flagging only potential threats with high levels of alert while running the algorithm for all moving targets in parallel, the operator can concentrate on evaluating the threats rather than searching for anomalous behaviour amongst a large number of targets, as shown in Fig. 5.29.

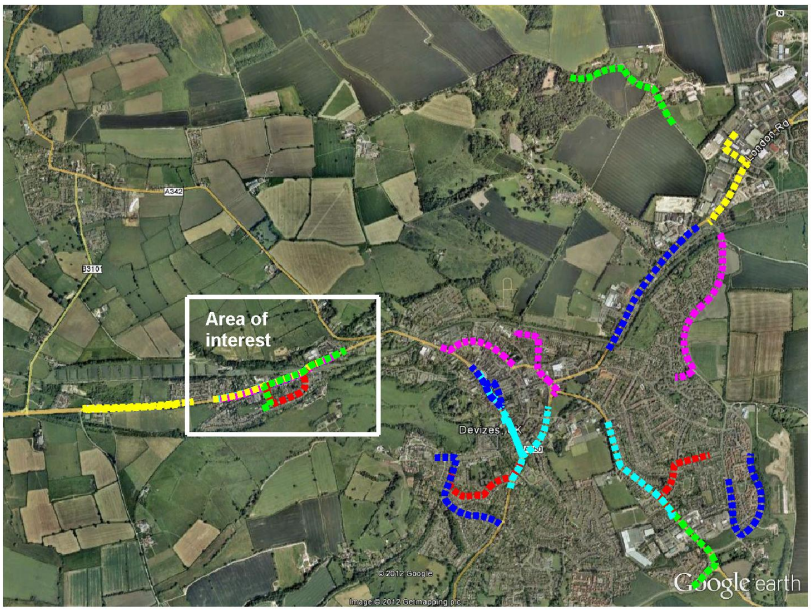


Figure 5.23: Trajectories of a ground vehicles within the Devizes road network with GIS satellite data overlaid thanks to Google earth[3]

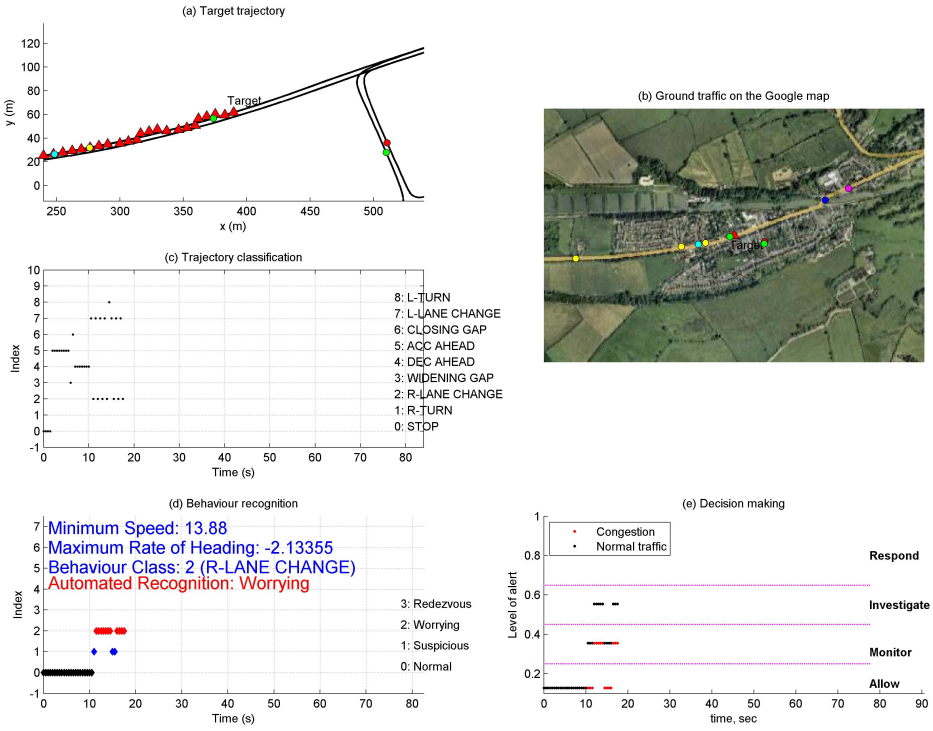


Figure 5.24: Behaviour recognition result of ground target 1 for civilian traffic scenario: evasive manoeuvring and rendezvous at 19 second

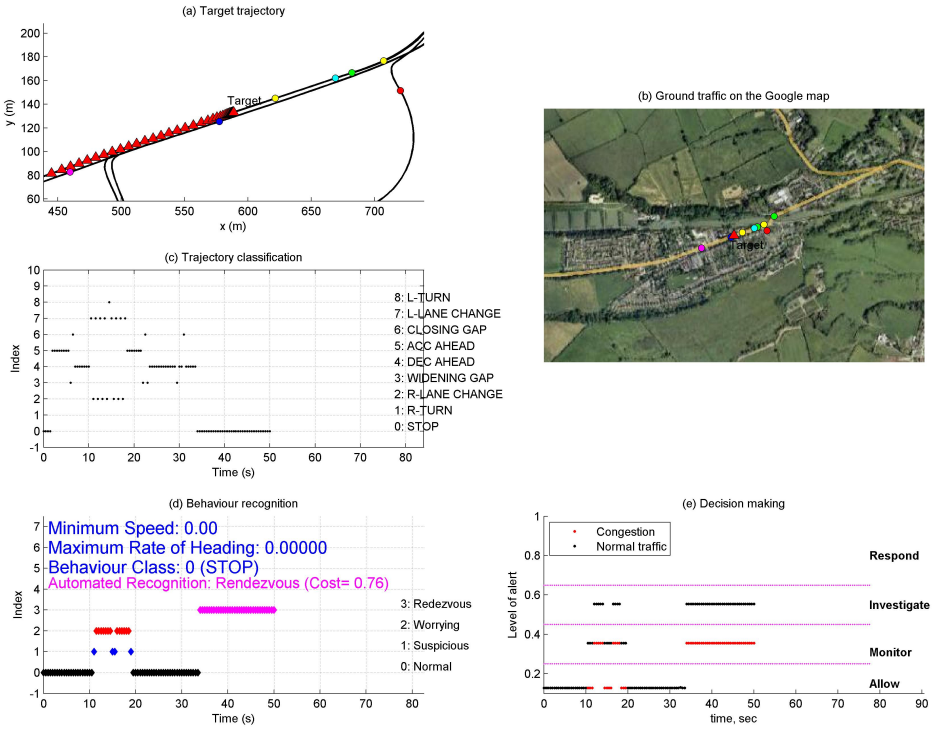


Figure 5.25: Behaviour recognition result of ground target 1 for civilian traffic scenario: evasive manoeuvring and rendezvous at 50 second

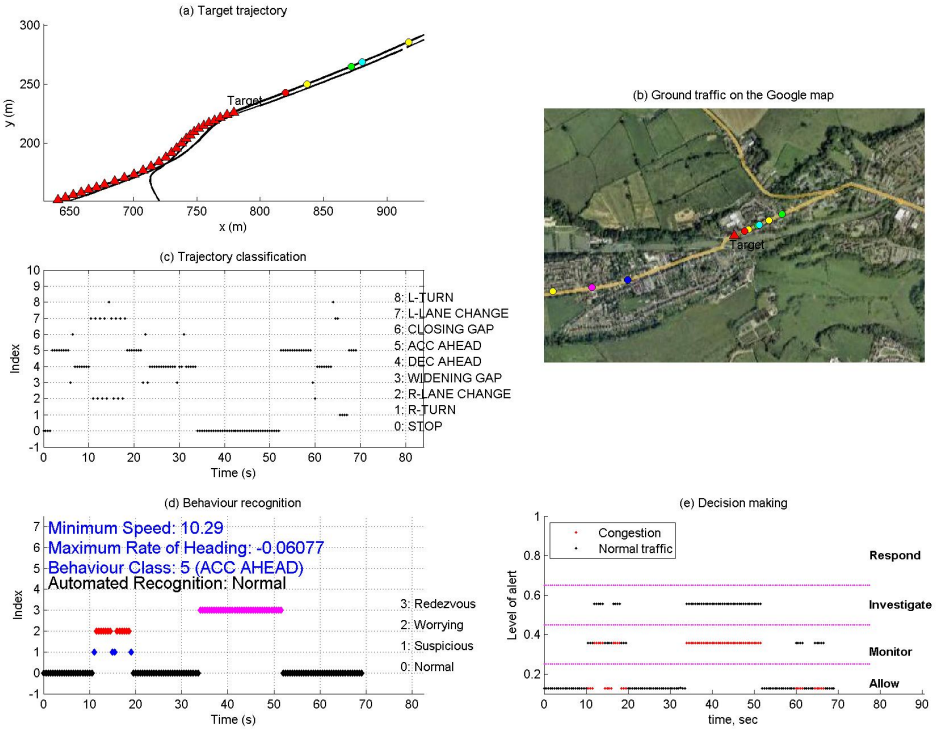


Figure 5.26: Behaviour recognition result of ground target 1 for civilian traffic scenario:evasive manoeuvring and rendezvous at the final time

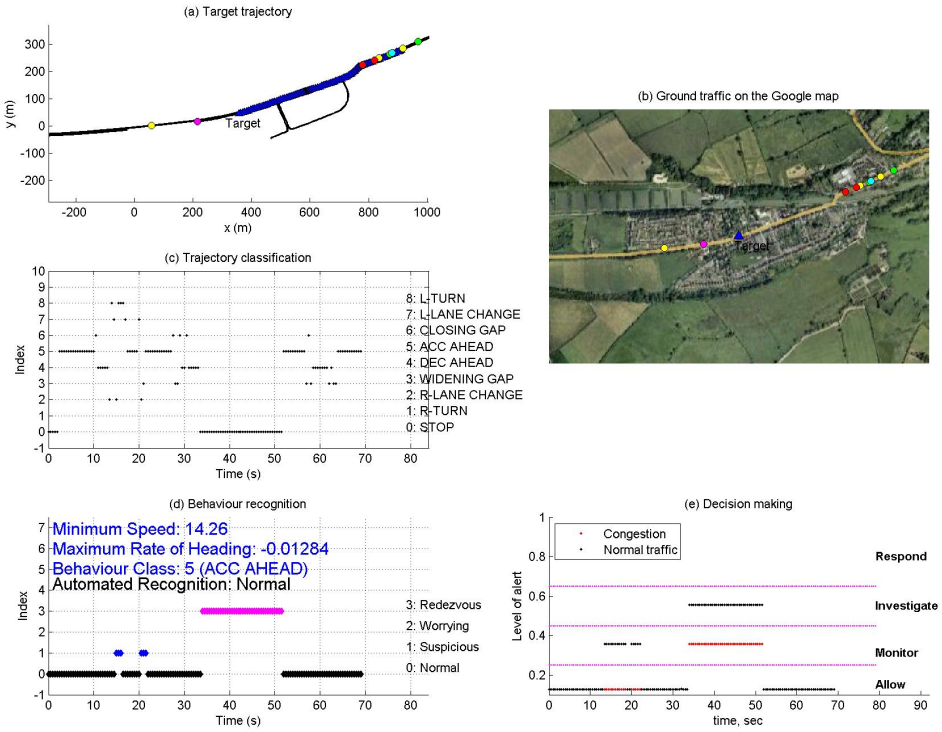


Figure 5.27: Behaviour recognition result of ground target 2 for civilian traffic scenario: rendezvous

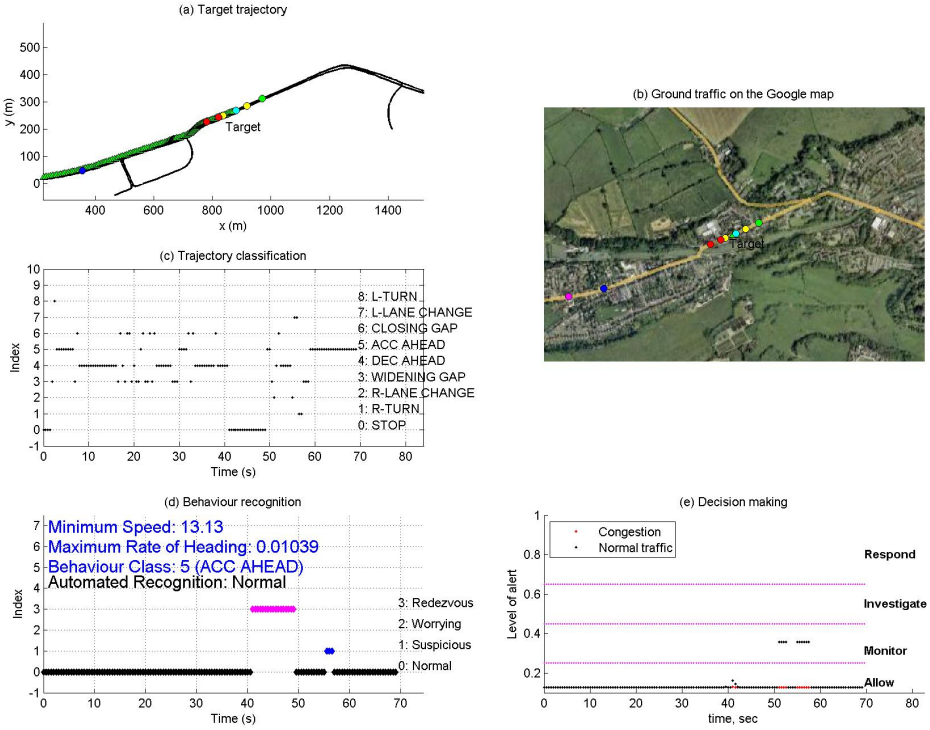


Figure 5.28: Behaviour recognition result of ground target 3 for civilian traffic scenario: rendezvous

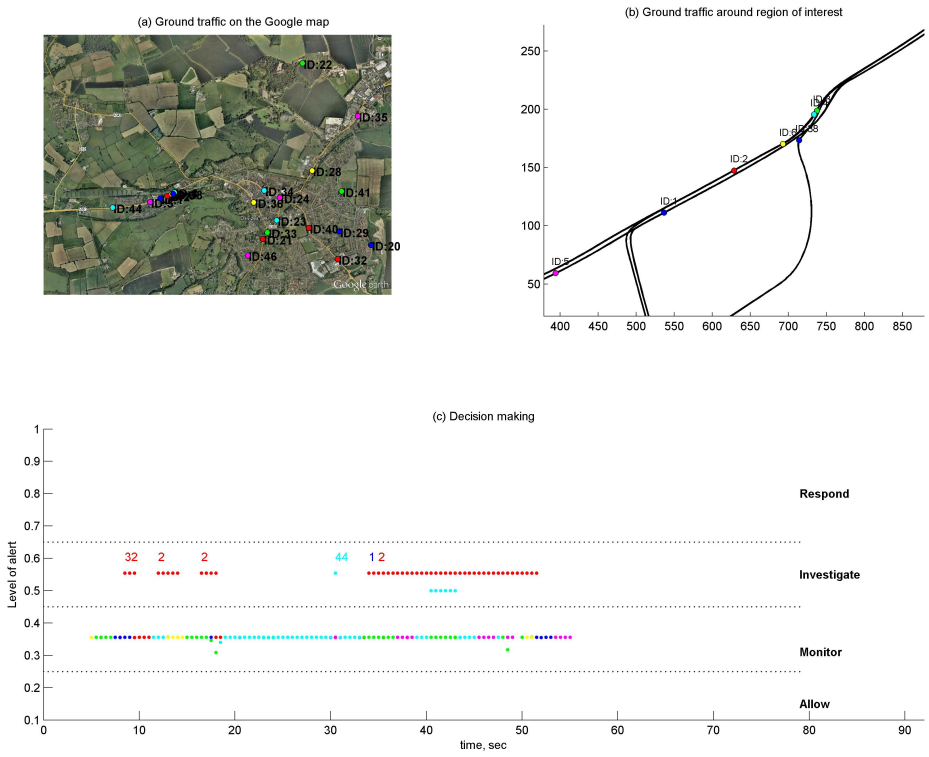


Figure 5.29: Behaviour recognition result of ground traffic with flagged potential threats

5.4 Summary

This chapter proposed an integrated behaviour recognition methodology for ground vehicles using UAVs, to identify suspicious or abnormal behaviour including the following techniques: a target tracking filter, sensor fusion, trajectory classification, behaviour detection and recognition. Numerical simulation results using real VMTI measurements demonstrated the feasibility and validity of the proposed approach. Furthermore, a synthetic military scenario and realistic car trajectory data from the off-the-shelf traffic S-Paramics simulation program were exploited to extend our approach to more general cases, by introducing a fuzzy rule-based decision making algorithm, as well as a Gaussian regression process. The proposed approach could be applied to various anomalous behaviour detection scenarios in support of Navy, Army, and Air Force providing decision making for ground or maritime traffic monitoring from the air. When integrated into an ISTAR (Intelligence, surveillance, target acquisition, and reconnaissance) system, this work will enable faster and more efficient situation awareness to be obtained and will significantly reduce the operator workload by indicating potential threats automatically for further monitoring. In the subsequent chapter, this further monitoring on the identified target will be addressed.

Chapter 6

Coordinated Standoff Tracking of a Ground Moving Target

Once a target is identified through airborne surveillance systems, further monitoring is required to obtain closer and higher-resolution surveillance data by approaching more closely and following it. In performing such missions, UAVs are to keep a certain distance from the moving target with a prescribed inter-vehicle angular separation amongst them, in order to track it without being noticed and, at the same time, to acquire accurate target information. The certain relative distance from the target is called the standoff distance, and therefore this approach is known as standoff target tracking. A configuration of UAVs for standoff tracking generally adopts two strategies: (i) distribute UAVs to have equal angular separation around the target [133] or (ii) position them at a certain angular position in the orbit. In the second strategy, the number of team members and sensor characteristics are taken into account to determine the angular separation maximising the estimation accuracy of target information [30, 134]. The requirement to maintain this configuration between UAVs, while tracking a moving target simultaneously, increases the complexity of the tracking guidance problem significantly.

For this standoff tracking problem, Lawrence [135] first proposed the application of Lyapunov vector fields for standoff coordination of multiple UAVs, which was further investigated by Frew *et al.* [14, 136] to include phase keeping as well as standoff tracking. They invented a decoupled control structure in which speed and rate of heading change are separately controlled for standoff distance and phase angle keeping, respectively. Summers *et al.* [137] extended this phase-keeping idea to multiple UAVs using information architectures in vehicle formations. Moreover, they addressed a variable airspeed controller to maintain the circular orbit despite unknown wind and unknown constant velocity target motion using adaptive estimation. Chen *et al.* [138] proposed the use of a tangent-plus-Lyapunov vector field which includes a simple switching logic between tangent and Lyapunov vector fields to make convergence to the standoff circle faster.

Note that, considering small allowable velocity bounds of the UAV and time delay in the velocity control, the convergence to the variable velocity command might be slow or hard to achieve. Thus, maintaining UAVs at their nominal and fuel efficient speed is desirable in terms of tracking performance, as well as from a mission duration point of view. In this regard, Kingston *et al.* [139] introduced a sliding mode control concept for circular formation and orbit radius change without velocity control for phase keeping of multiple UAVs. Sepulchre *et al.* [140] and Paley *et al.* [141] introduced the notion of the splay state which represents uniform rotation of evenly spaced vehicles on a circle. They applied the collective control of multi-agent system to stabilse symmetric circular formation around the target using unit speed vehicles. Klein and Morgansen [142] proposed a steering control law making the velocity of the collective centroid match a reference velocity allowing tracking of a moving target. Note also that policies that maintain the optimal standoff distance while allowing the angular separation to vary can be better in terms of estimation accuracy than those that maintain optimal angular separation with orbit radius change while compromising the standoff distance [30]. Accordingly, a tracking guidance algorithm that explicitly exploits both orbit radius and velocity control concept is required to be developed within a unified framework.

Wise and Rysdyk [29] surveyed and compared the different methodologies for standoff tracking: these were the Helmsman behavior, Lyapunov vector field, controlled collective motion, and model predictive control. Prevost *et al.* [143] applied a receding horizon model-based predictive control for standoff tracking. Although the aforementioned two works tried to apply the model predictive control (also known as receding horizon control), Wise and Rysdyk [29] used only a simple relative kinematics variation between the UAV and the target while still decoupling the speed and rate of heading change, and Prevost *et al.* [143] focused on single UAV applications for only standoff distance keeping with a simple UAV/target trajectory prediction.

Note that the performance of the tracking guidance algorithms heavily depends on the sensing and estimation capability to determine target's position and intent. Thus, sensor fusion using other UAV's measurements and estimates for the same target along with contextual information could enhance the performance of the guidance as well as the estimation. Especially, in many applications for the

ground target tracking, the majority of ground vehicles are moving on road networks whose topographical coordinates could be known with a certain accuracy. Such road-map information can be used for improving the estimation accuracy, by constraining the state of the ground target in its position and velocity within the road geometry, as explained in Chapter 4. However, the road-constrained estimation has rarely been combined with standoff tracking guidance to our best knowledge, even though ground vehicle of interest is moving only on the road in many cases.

With these backgrounds, this chapter proposes four coordinated standoff tracking algorithms, some of which is in conjunction with a road-map assisted filtering and sensor fusion techniques, as follows:

1. **Two-phase orbit approach based on path shaping;**

   The path shaping approach is proposed for stationary target tracking without velocity change. A simultaneous arrival concept is first introduced as a coordinated tracking strategy for multiple UAVs, which initialises UAVs on a standoff orbit with a desired angular separation. In order to address arrival time delay or failure of the UAV, a two-orbit approach is newly proposed in which UAVs first arrive at the outer orbit and subsequently shrink to the desired inner orbit at different time, while adjusting the angular separation between constant-speed UAVs.

2. **Decentralised vector field guidance using adaptive sliding mode control;**

   To cope with an unknown moving ground target, a tangent vector field guidance algorithm using sliding mode control (SMC) is applied. This study introduces adaptive term in the SMC approach to reduce the effect of unmodelled dynamics and disturbance in the heading-hold autopilot. Decentralised angular separation guidance along with decentralised estimation is also proposed using either velocity change or orbit radius change by different communication/information architectures.

3. **Standoff tracking guidance based on differential geometry;** and

   Then, a rendezvous and standoff tracking guidance using differential geometry concepts is proposed. The proposed algorithm brings several advantages along with its inherent simplicity: rigorous stability and explicit use

of a target velocity. Moreover, the proposed guidance law requires the reduced number of tuning variables, only a curvature command, unlike other approaches such as vector field guidance requiring appropriate vector field generation as well as guiding vehicle into the field.

4. **Nonlinear model predictive coordinated standoff tracking.**

   Lastly, this chapter proposes an NMPCST (Nonlinear Model Predictive Coordinated Standoff Tracking) framework for both standoff distance and phase angle keeping by a pair UAVs. Unlike a decoupled control structure of the vector field based works, this coupled approach provides an optimal performance in terms of tracking accuracy as well as control efforts. Besides, in this sort of model predictive control, since the prediction of the target movement plays a more important role on the tracking performance, this study investigates the effect of improved estimation accuracy on the tracking guidance performance.

6.1 Path Shaping Approach

Unlike the existing approaches focusing on the guidance law design changing turn rate and speed, this section describes a novel path shaping strategy taking kinematic constraints of multiple UAVs into account for a coordinated standoff target tracking. Since fixed-wing UAVs fly at a nominal airspeed for a longer duration with better fuel efficiency than a rotary-wing UAV with hovering capability, it is desirable that they keep angular separation between vehicles while holding a constant velocity. In order to produce flyable paths to fit to the fixed-wing UAV dynamics, the path shaping algorithm needs to consider the most critical constraint of fixed-wing UAVs: curvature of turning maneuver, which is dependent on the operating range of speed and bank angle. Moreover, it is advantageous for the shaped trajectories to have a shorter length as well as lower curvature differences at discontinuous points as possible in order to minimize flight time or energy consumption [45]. For this, two constant curvature segments is exploited explained in Section 2.2 for path shaping. As a measure to coordinated target tracking for multiple UAVs, a simultaneous arrival concept is first introduced, which initialises UAVs on a standoff orbit with a desired angular separation. In

order to address arrival time delay or failure of the UAV, a two-orbit approach is newly proposed in which UAVs first arrive at the outer orbit and subsequently shrink to the desired inner orbit at different time, while adjusting the angular separation between constant-speed UAVs.

As a main contribution, the proposed standoff tracking strategy is more realistic because it is based on the constant-speed maneuvers of fixed-wing UAVs, whereas most of the previous research considers speed variations for phase separation between the UAVs. Another contribution is that using path shaping for standoff tracking can consider constraints of UAV kinematics (i.e. curvature or turn radius) inherently, generating feasible paths for UAVs at all times, whereas the existing literature often needs to limit a turn rate or curvature command from the guidance loop to the saturation values. Note that this study considers the path shaping approach using kinematic constraints of the UAV, providing an open-loop path. For real implementation, feedback control to follow the generated path is required to cope with gust and disturbances. However, the proposed approach still has an advantage over the other guidance algorithms which require sharing of information continuously between UAVs for simultaneous arrival. Meanwhile, in the proposed two-orbit approach, UAVs need to follow the generated path and communicate each other only twice when each UAV reaches the outer orbit. To verify the feasibility and benefits of the proposed coordinated standoff tracking strategy, numerical simulations are performed for both stationary and moving ground targets using two cooperative UAVs.

6.1.1 Simultaneous Arrival

Given initial positions and orientations for UAVs and target information, it is possible to determine two arrival points satisfying the standoff distance as well as the desired angular separation between UAVs in terms of the line of sight angle relative to the target. However, such a goal can be satisfied fully only if the two vehicles reach the respective position at the same time, while the path shaping method does not provide any control on the arrival time directly. Assuming the constant velocity of the vehicle, the time required to get to the designated position proportional to the path length with the speed. Thus, if there are no errors and disturbances in the system, the arrival time can be determined

explicitly by adjusting the curvature of the path. In path shaping using constant curvature segments, the arc length is simply obtained by using the curvature κ and the turning angle θ as:

$$l_a = \frac{\theta}{\kappa}. \tag{6.1}$$

By using the formulas given in Section 2.2, the path length l_{sf} between initial and final positions can be expressed as:

$$l_{sf} = \frac{d}{2}\left[\frac{\sin(\phi_f - \gamma_f)}{\sin\phi_s}\theta_s + \frac{\sin(\phi_s - \gamma_s)}{\sin\phi_f}\theta_f\right] \tag{6.2}$$

where ϕ_s is the only independent variable, which can be used as a tuning parameter for the path length. Noting that $(\phi_f, \theta_s, \theta_f)$ vary almost linearly with ϕ_s, and its valid range is usually not too wide because of the physical constraints, the path length function l_{sf} has a nearly-linear behavior. Defining squared path length difference as a performance index, optimal feasible solution ϕ_s for two UAVs can be obtained solving a constrained nonlinear minimization problem, given as:

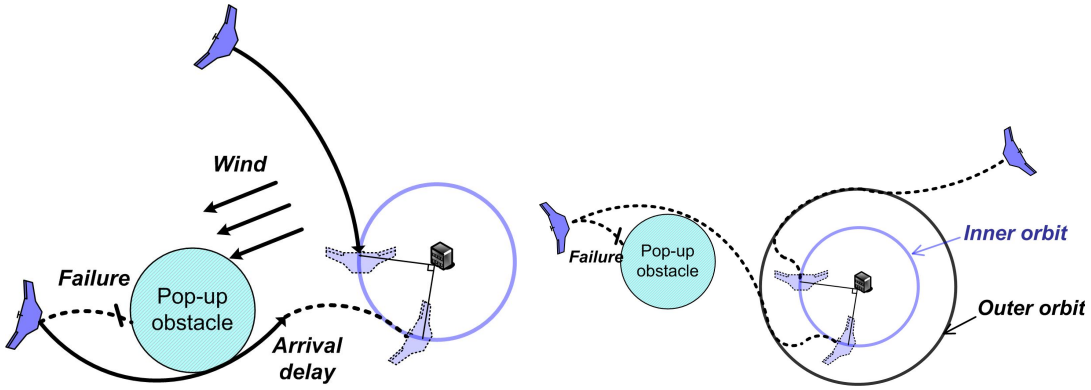
$$(\phi_{s_1}^*,\ \phi_{s_2}^*) = arg \min_{(\phi_{s_1},\ \phi_{s_2})} (l_{sf_1} - l_{sf_2})^2 \tag{6.3}$$

where subscript represents value for the i-th UAV. In case that path length ranges for the two vehicles partially overlap, there are many solutions providing the same path length (i.e. same arrival time). Depending on the task objectives, either the shortest path solution or the one requiring the smallest curvature variation can be chosen.

6.1.2 Phase Correction by a Two-Orbit Approach

When initial positions of two UAVs are far away, the simultaneous arrival is not possible to achieve, or demands too much energy due to keep the time constraint resulting in the long detour of one of the UAVs. Moreover, to minimise the exposure time of UAVs in the enemy radar site while maximizing the effect of simultaneous arrival to the designated position, the mission planner should be able to cope with the failure of the UAV or the arrival time delay by wind or unexpected obstacles during the mission as illustrated in Fig. 6.1(a). To address aforementioned problems, a two-orbit approach for the phase control is introduced as shown in Fig. 6.1(b). The idea is to reach the observation (or outer) orbit

first, and then to perform a further (referred in the following as 'shrink') manoeuvre identical for two UAVs that brings the UAVs on an inner orbit at different time. The difference of time spent on the outer orbit allows to adjust the angular separation of the UAVs.



(a) Simultaneous arrival (b) Two-orbit approach

Figure 6.1: Motivation of a two-orbit approach

Once one of UAVs reaches the outer orbit, that UAV remains on the same orbit until the other UAV reaches the orbit. Let us define the time t_r as the time both UAVs get on the outer orbit and leader and follower based on the relative phase angle with respect to the target as given:

$$Leader = arg \max_{UAV_i} (\sigma_{UAV_1}(t_r), \sigma_{UAV_2}(t_r)) \tag{6.4}$$

where $\sigma_{UAV_i} \in [0, 2\pi)$ represents the phase angle of the i-th UAV. The shrink manoeuvre is illustrated as shown in Fig. 6.2. Given the desired standoff distance which is equivalent to the inner orbit radius r_i, parameters to be chosen are an outer orbit radius r_o and an approach angle ν. Then, the arrival point P_2 on the inner orbit and the orientation angle are uniquely defined given these values, and the path shaping technique using constant curvature segments is applied to connect configuration P_1 and P_2. It is worth noting that since the shrink manoeuvre itself represents a sort of an additional path, ν should be designed to make the path as short as possible. Moreover, the smaller value of r_i closer to the outer orbit radius will lead to a smaller time needed for the shrink manoeuvre, however, on the other hand, it will take more time to achieve the desired angular separation between UAVs; parameter r_o and ν should be determined considering a trade-off between these two aspects. To achieve the desired angular separation, let us define the phase difference between UAVs at time t_r as:

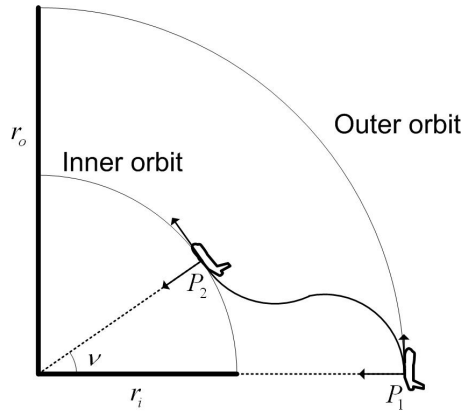
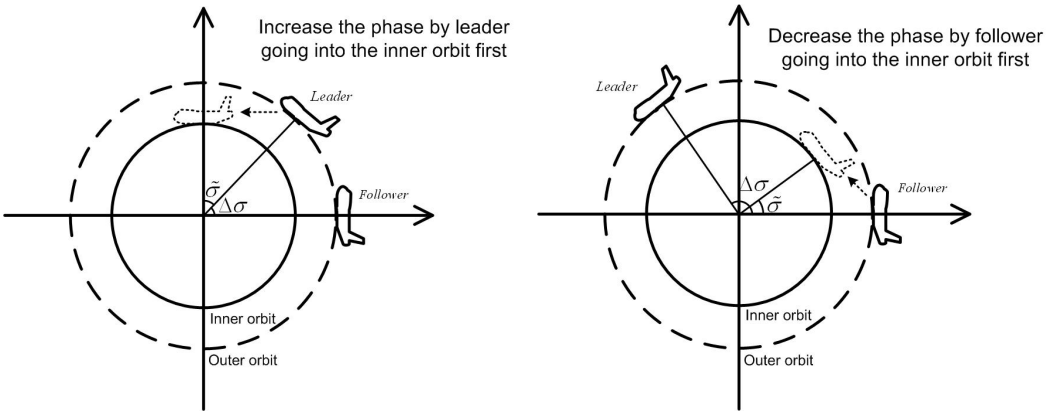


Figure 6.2: An illustration of the two-orbit approach



(a) Leader entering the inner orbit first *(b) Follower entering the inner orbit first*

Figure 6.3: Phase correction by one of UAVs depending on the situation

$$\begin{aligned} \Delta\sigma &= \sigma_l - \sigma_f, \qquad \text{if } \sigma_l - \sigma_f \leq \pi \\ &= \sigma_l - \sigma_f - 2\pi, \quad \text{otherwise} \end{aligned} \tag{6.5}$$

where σ_l and σ_f represent the phase angle of leader and follower at time t_r with respect to the target, respectively. Then, the phase angle error $\tilde{\sigma}$ which needs to be corrected by the two-orbit approach is defined as:

$$\tilde{\sigma} = (\sigma_d - |\Delta\sigma|)\text{sign}(\Delta\sigma) \tag{6.6}$$

where σ_d is the desired angular separation. If the phase angle error $\tilde{\sigma}$ is positive, the leader should enter the inner orbit first to increase the angular separation from $\Delta\sigma$ to the σ_d. Similarly, if $\tilde{\sigma}$ is negative, the follower will shrink to the inner orbit first to decrease the angular separation. Figure 6.3 shows an exemplary decision as to the UAV entering the inner orbit first in case of $\sigma_d = \frac{\pi}{2}$. Once one

appropriate UAV reaches the inner orbit by the shrink manoeuvre, the other UAV needs to wait for a certain amount of time before starting the shrink manoeuvre. Considering the UAV flying on the orbit of radius r with a constant speed V, the angular variation α during time t is given:

$$\alpha = \frac{Vt}{r}. \tag{6.7}$$

Thus, time T_c needed to correct the phase difference $\triangle\sigma$ to the desired value by the two different orbit can be obtained as:

$$\begin{aligned} |\tilde{\sigma}| &= \frac{VT_c}{r_i} - \frac{VT_c}{r_o} \\ T_c &= \frac{|\tilde{\sigma}| r_o r_i}{V(r_o - r_i)}. \end{aligned} \tag{6.8}$$

The aforementioned path shaping algorithm is for the stationary target in general. However, it can be extended to the moving target case in a certain situation such that the update rate of the target information is very low, expected future information for a lost target is given, or target movement can be predicted precisely. In those situations, the objective of the path shaping could be the gathering of UAVs to the appropriate position on the standoff orbit with the desired angular separation considering target movement. For this, let us first define the maximum time for the phase angle correction $T_{c_{max}}$ with the relation $|\tilde{\sigma}| \leq \sigma_d$ from Eq. (6.6) as:

$$T_{c_{max}} = \frac{|\tilde{\sigma}|_{max} r_o r_i}{V(r_o - r_i)} = \frac{\sigma_d r_o r_i}{V(r_o - r_i)}. \tag{6.9}$$

When both UAVs reach the outer orbit of the expected future target position at time t_d, the remaining time t_{rem} should be greater than the time needed for the shrink manoeuvre of both UAVs and the phase angle correction $T_{c_{max}}$ as given:

$$t_{rem} = t_d - t_a \geq 2t_{sh} + T_{c_{max}} \tag{6.10}$$

where t_a represents the latest arrival flight time of UAVs to the outer orbit from the initial positions, and t_{sh} represents the time needed for shrink manoeuvre which can be obtained in advance by predetermined values of r_o, r_i and v. Then, minimum time t_d to gather UAVs around the expected target position at that

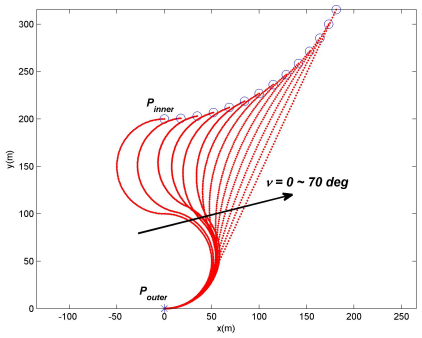
time while keeping the desired standoff distance and the angular separation is computed as:

$$\begin{aligned} &\min\ t_d \\ &\text{s.t}\ t_d \ \geq\ t_a + 2t_{sh} + T_{c_{max}} \end{aligned} \tag{6.11}$$

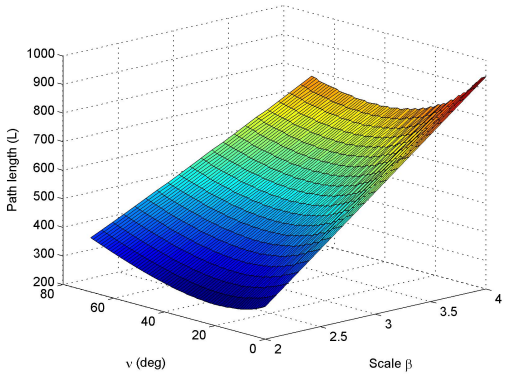
Once cooperative UAVs achieve the goal (the standoff distance and the angular separation) using the path shaping approach, an active guidance algorithm such as the vector field guidance [14, 139] can be applied to track the manoeuvring target tightly while estimating information of the target through filtering as in [96, 144].

6.1.3 Numerical Simulations

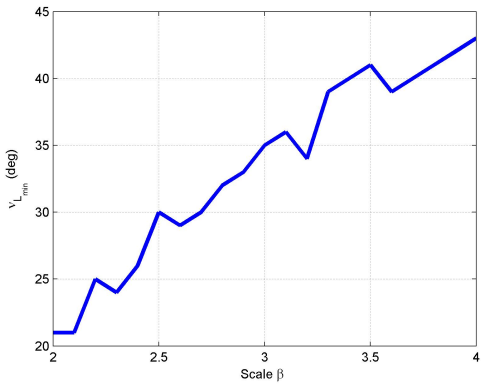
To apply the two-orbit approach, given the constant velocity of 40 m/s and the desired inner orbit radius of 200 m, the appropriate value for the shrinking manoeuvre is investigated with different approach angle v and orbit radius ratio $\beta = r_o/r_i$ as shown in Fig. 6.4. As explained earlier, parameter β and v should be determined considering a trade-off between two aspects, which are the additional path length due to the shrink manoeuvre and the time required for the phase correction. To get a reasonable phase correction time, the orbit radius ratio β can be suggested to be greater than two as shown Fig. 6.4(d), and then v can be determined accordingly as shown in Fig. 6.4(c). Figure 6.5 shows the path shaping result for the coordinated standoff tracking of the stationary target with $V = 40$ m/s, $r_i = 200$ m, $\beta = 2$, $v = 20^\circ$ and $\sigma_d = \frac{\pi}{2}$. At time t_r, both of UAVs have reached the outer orbit, and then the follower (UAV_1 which reaches the outer orbit later) first enters into the inner orbit to decrease the phase angle difference between UAVs to a desired value σ_d. After T_c seconds, the leader goes into the inner orbit using the same shrink manoeuvre as that of the follower. At 34 seconds, both the standoff distance ($r_i = 200$ m) from the target and desired angular separation ($\sigma_d = \frac{\pi}{2}$) between two UAVs are satisfied at the same time as shown in Fig. 6.5. The path shaping result for the coordinated standoff tracking of the moving target is shown in Fig. 6.6. The speed of UAV is set to 40 m/s and target 10 m/s. Minimum time t_d to gather UAVs around the target is computed firstly, and the similar approach as the stationary target case is applied with the expected position of target



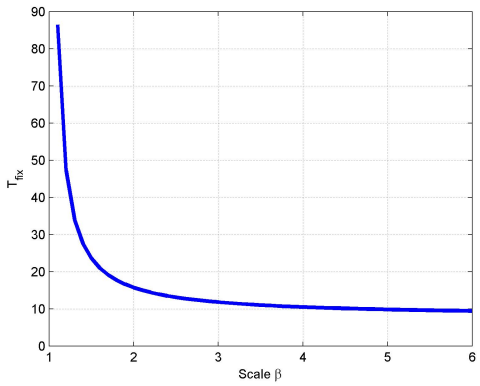
(a) *Example trajectory of shrink manoeuvre* ($\beta = \frac{r_o}{r_c} = 2$)



(b) *Path length*



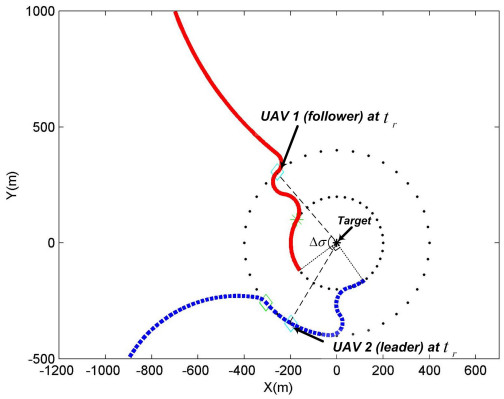
(c) *Approach angle ν having minimum path length*



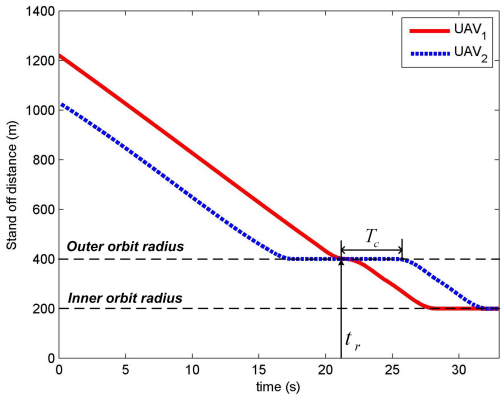
(d) *Time for the phase correction* ($\tilde{\sigma} = \frac{\pi}{2}$)

Figure 6.4: Effect on shrink manoeuvre of different parameters ($V = 40m/s, r_i = 200m$)

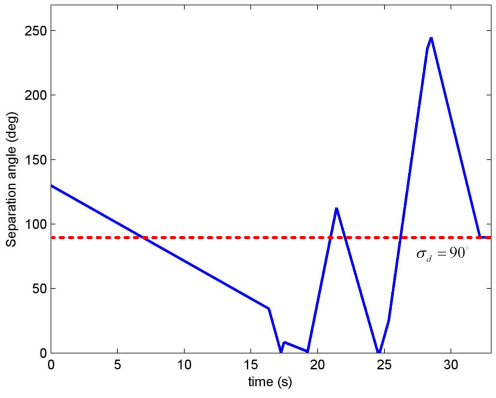
at that time. At 53 seconds, both UAVs are gathered while satisfying the desired standoff distance from the target and angular separation between them.



(a) Trajectory of two UAVs

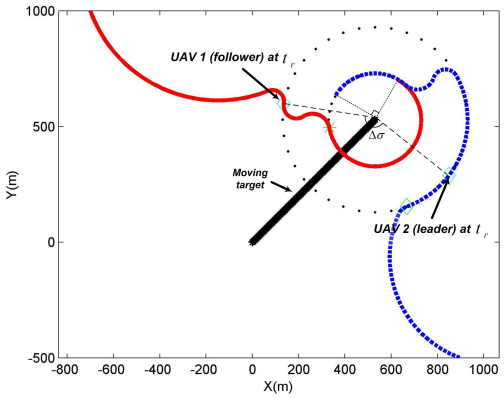


(b) Standoff distance from the target

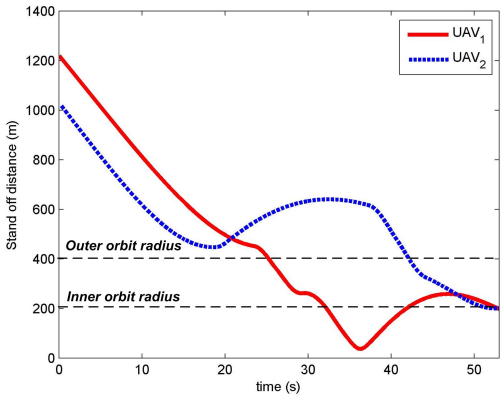


(c) Angular separation between two UAVs

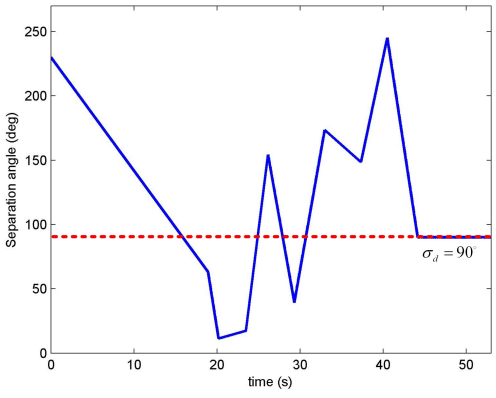
Figure 6.5: Path shaping result for coordinated standoff tracking of the stationary target



(a) Trajectory of two UAVs



(b) Standoff distance from the target



(c) Angular separation between two UAVs

Figure 6.6: Path shaping result for coordinated standoff tracking of the moving target

6.2 Decentralised Vector Field Guidance with Adaptive SMC

This section proposes coordinated standoff tracking guidance algorithms of a ground moving target using sliding model control (SMC), based on vector field approach. Motivated by the aforementioned work [139], this study first exploits additional adaptive terms in the existing vector field approach using sliding mode control, in order to reduce the effect of unmodeled dynamics and disturbances in the heading-hold autopilot. Then, decentralised angular separation control for multiple UAVs is introduced using either velocity or orbit radius change with a constant airspeed by different information structures, following [137, 139]. Note that, it is hard to have a centralised system that handles entire information and controls for a team of UAVs considering communication and computation constraints. Therefore, control laws are decentralised using local information about a target and neighbouring UAVs. Moreover, in order to obtain accurate position and motion information of a moving target to be used for UAV guidance, as well as to cope with the proposed decentralised guidance approach, this study applies the decentralised sensor fusion approach. Numerical simulations are extensively performed to verify the feasibility and compare benefits of the proposed guidance algorithms, under a realistic ground vehicle tracking scenario, using multiple UAVs having unknown parameters in the system.

6.2.1 Vector Field Characteristic

This section briefly reviews several recent vector field guidance approaches for standoff tracking of a ground moving target, including their formulation and numerical example results for stationary target tracking.

6.2.1.1 System dynamics

Assuming each UAV has a low-level flight controller such as SAS (Stability Augmentation System) and CAS (Controllability Augmentation System) for heading and velocity hold functions, this study aims to design guidance inputs to this

low-level controller for standoff target tracking. Consider a two-dimensional UAV kinematic model [96] as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{v} \\ \dot{\omega} \end{pmatrix} = f(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} v\cos\psi \\ v\sin\psi \\ \omega \\ -\frac{1}{\tau_v}v + \frac{1}{\tau_v}u_v \\ -\frac{1}{\tau_\omega}\omega + \frac{1}{\tau_\omega}u_\omega \end{pmatrix} \tag{6.12}$$

where $\mathbf{x} = (x, y, \psi, v, \omega)^T$ are the inertial position, heading, speed and yaw rate of the UAV, respectively. τ_v and τ_ω are time constants for considering actuator delay. $\mathbf{u} = (u_v, u_\omega)^T$ are the commanded speed and turning rate constrained by the following dynamic limits of a fixed-wing UAV:

$$|u_v - v_0| \leq v_{max} \tag{6.13}$$

$$|u_\omega| \leq \omega_{max} \tag{6.14}$$

where v_0 is a nominal speed of UAV. The continuous UAV model in Eq. (6.12) can be discretised by Euler integration into:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, \mathbf{u}_k) \tag{6.15}$$

where $\mathbf{x}_k = (x_k, y_k, \psi_k, v_k, \omega_k)^T$, $\mathbf{u}_k = (u_{vk}, u_{\omega k})^T$, and T_s is a sampling time. If the frequencies of the tracking guidance and autopilot are not too close, it is common to initially design and verify the guidance law and control algorithm separately. Therefore, like in many literatures considering similar guidance problems [14, 96, 139], above simple kinematics is used for the UAV model. However, the final validation needs to be made with higher complexity simulation models and flight tests, and these remain as future work.

6.2.1.2 Lyapunov vector field

Let us consider the Lyapunov vector field first, which was initially proposed by Lawrence [135] and further developed by Frew *et al.* [14] as:

$$V(x, y) = (r^2 - r_d^2)^2 \tag{6.16}$$

where $r = \sqrt{\delta x^2 + \delta y^2} = \sqrt{(x - x_t)^2 + (y - y_t)^2}$ is the distance of the UAV from the ground vehicle. Herein (x_t, y_t) is the position of the ground vehicle which can be estimated from the tracking filter, and r_d is a desired standoff distance from the UAV to the ground vehicle. Differentiating Eq. (6.16) gives:

$$\dot{V}(x, y) = \nabla V[\dot{x}, \dot{y}]^T. \tag{6.17}$$

The Lyapunov vector field uses the following desired velocity $[\dot{x}_d, \dot{y}_d]^T$:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \frac{-V_d}{r(r^2 + r_d^2)} \begin{bmatrix} \frac{\delta x}{k_l}(r^2 - r_d^2) + \delta y(2rr_d) \\ \frac{\delta y}{k_l}(r^2 - r_d^2) - \delta x(2rr_d) \end{bmatrix} \tag{6.18}$$

where $0 < k_l \leq 1$ is positive constant, and V_d is a desired UAV speed. Substituting Eq. (6.18) into Eq. (6.17) yields

$$\dot{V}(x, y) = -\frac{4V_d r}{k_l(r^2 + r_d^2)} V. \tag{6.19}$$

In Eq. (6.19), $\dot{V} \leq 0$ except $r = 0$, and then r converges to the largest invariant set $r = r_d$ satisfying $\dot{V} = 0$ by LaSalles's invariance principle [145]. Note that k_l, newly introduced in this study, can be used to adjust the converging speed of the generated field to the standoff circle. The desired heading can be decided using the desired two dimensional velocity components in Eq. (6.18) as:

$$\psi_d = \tan^{-1} \frac{\dot{y}_d}{\dot{x}_d}. \tag{6.20}$$

The guidance command u_ψ for turn rate is selected as the sum of proportional feedback and feedforward terms as:

$$u_\omega = -k_\psi(\psi - \psi_d) + \dot{\psi}_d \tag{6.21}$$

where $\dot{\psi}_d = 4v_d \frac{r_d r^2}{(r^2 + r_d^2)^2}$ can be obtained by differentiating Eq. (6.20).

6.2.1.3 Supercritical Hopf bifurcation

The second vector field is the supercritical Hopf bifurcation which was initially proposed in [146] based on the theory of [147]. The Supercritical Hopf bifurcation is known to mathematically produce the spiral trajectory which converges to a

limit cycle; no matter where the starting position is located. This property is similar to that of the Lyapunov vector field. Let us consider the following desired vector field based on the system equation of the Supercritical Hopf bifurcation:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \begin{bmatrix} \frac{\delta x}{k_s r_d^2}(r_d^2 - \delta x^2 - \delta y^2) - \delta y \\ \frac{\delta y}{k_s r_d^2}(r_d^2 - \delta x^2 - \delta y^2 + \delta x \end{bmatrix} \tag{6.22}$$

where $0 < k_s \leq 1$ is a positive constant, and the definitions of the other variables are the same as those of the Lyapunov vector field. Substituting Eq. (6.22) into Eq. (6.17) yields

$$\dot{V}(x, y) = -\frac{4r}{k_s r_d^2} V \tag{6.23}$$

where k_s can be used to adjust the converging speed to the standoff circle, and in Eq. (6.23), $\dot{V} \leq 0$ except $r = 0$, and then r converges to the largest invariant set $r = r_d$ satisfying $\dot{V} = 0$ by LaSalles's invariance principle [145]. The desired heading angle and guidance command are determined in a similar way as Eqs. (6.20)~(6.21).

6.2.1.4 Tangent vector field

The tangent vector field is generated by a desired heading angle as shown in Fig. 6.7, given by [139]:

$$\psi_d = \psi_p + \tan^{-1}(k_t d) \tag{6.24}$$

where $d = r - r_d$ is the distance of the UAV from the desired standoff orbit, and ψ_p is tangent to the standoff orbit along the ray connecting the UAV and the target position as:

$$\psi_p = \theta + \frac{\pi}{2} \tag{6.25}$$

where $\theta = \tan^{-1}(\delta y / \delta x)$.

To analyse the stability of guidance algorithm, vehicle dynamics following a desired heading angle is expressed in polar coordinates, by differentiating r and θ as:

$$\begin{aligned} \dot{r} &= -V_d \sin\tilde{\psi} \\ \dot{\theta} &= \frac{V_d}{r} \cos\tilde{\psi} \end{aligned} \tag{6.26}$$

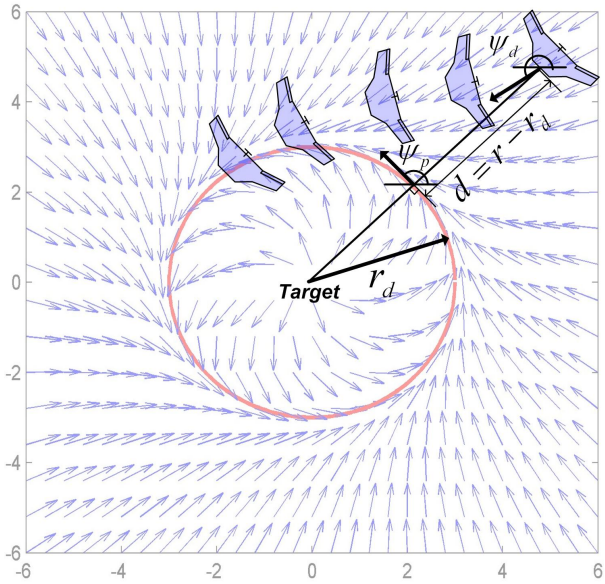


Figure 6.7: Geometry of tangent vector field

where $\tilde{\psi} = \psi_d - \psi_p$. Now, Lyapunov function is defined as $W = \frac{1}{2}\tilde{\psi}^2$. Differentiating this with Eq. (6.26) and $\tilde{\psi} \in (-\pi/2, \pi/2)$ gives:

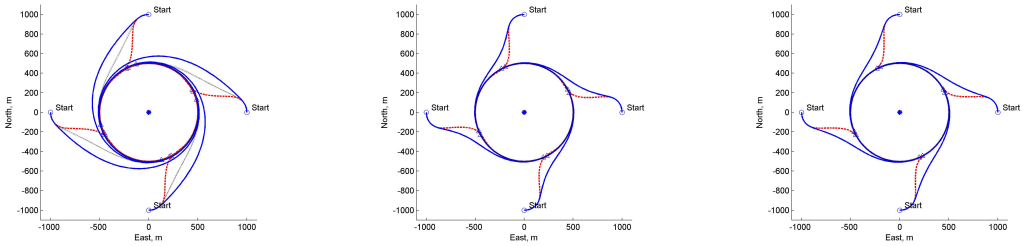
$$\dot{W} = \tilde{\psi}\dot{\tilde{\psi}} = -\frac{k_t V_d \tilde{\psi} \sin\tilde{\psi}}{1 + (k_t d)^2} < 0. \tag{6.27}$$

This shows that a heading angle converges to an angle ψ_p tangent to the orbit, and d consequently converges zero (that is, $r \to r_d$), asymptotically by LaSalle's invariance principle. Moreover, the feed-forward term $\dot{\psi}_d$ for the guidance command can be obtained by differentiating Eq. (6.24) as:

$$\dot{\psi}_d = \frac{V_d}{r}\cos\tilde{\psi} - \frac{k_t V_d}{1 + (k_t d)^2}\sin\tilde{\psi} \tag{6.28}$$

Figure 6.8 shows the example of UAV tracking trajectories for a stationary target using different vector field approaches. Although aforementioned three vector fields have different characteristics, they can generate a similar trajectory as red dashed lines in Fig. 6.8, by adjusting the field gain such as k_l, k_s and k_t. It is worth noting that the field gain should be determined not to exceed the turning rate constraint of the vehicle, ω_{max}, along with a control gain. This study uses a tangent vector field approach since it allows to easily apply a sliding mode control

concept, as well as the orbit radius change for angular separation control between UAVs.



(a) Lyapunov vector field *(b) Supercritical Hopf bifurcation field* *(c) Tangent vector field*

Figure 6.8: Example UAV trajectories for a stationary ground target using vector field approaches with different field gains

6.2.2 Tangent Vector Field Guidance With Adaptive SMC

6.2.2.1 Conventional and sliding mode control concept

This section presents the tangent vector field guidance (TVFG) strategy using a sliding mode control concept. First, the UAV is assumed to follow a first-order heading (or course) dynamics by the autopilot, as given:

$$\dot{\psi} = \alpha(\psi_c - \psi) + \nu \tag{6.29}$$

where ψ_c is the commanded heading, α is a known positive constant that characterises the speed of response of heading-hold autopilot loop, and ν represents unmodelled dynamics or disturbances of the autopilot loop. Let us first consider a conventional proportional-derivative (PD) type controller. If α were accurately known and ν were small enough to ignore, the guidance command ψ_c, to obtain the desired heading ψ_d in Eq. (6.24), would be selected as the sum of proportional feedback and feedforward term as:

$$\psi_c = \psi + \frac{1}{\alpha}(\dot{\psi}_d - k_\psi(\psi - \psi_d)) \tag{6.30}$$

where k_ψ represents a proportional gain.

However, in general, α is difficult to determine experimentally and inevitably contains error in its estimated value, and ν can have a considerable value due to

unmodeled dynamics, disturbances or faults. Since these factors could result in less precise target tracking, this study uses adaptive sliding mode control, which estimates and compensates the effect of unknown parameters.

First of all, let us define a sliding surface as:

$$S = e + k_I \int_0^t e d\tau \tag{6.31}$$

where $e = \psi - \psi_d$ is a tracking error, and k_I is an integral gain. Differentiating the sliding surface with respect to time gives:

$$\dot{S} = \dot{\psi} - \dot{\psi}_d + k_I e = \alpha(\psi_c - \psi) + \nu - \dot{\psi}_d + k_I e. \tag{6.32}$$

Consider a Lyapunov function candidate: $W_1 = \frac{1}{2}S^2$, and take the derivative to obtain:

$$\dot{W}_1 = S\dot{S} = S\left(\alpha(\psi_c - \psi) + \nu - \dot{\psi}_d + k_I e\right) \tag{6.33}$$

Then, the control command can be selected as:

$$\psi_c = \psi + \frac{1}{\hat{\alpha}}(-\hat{\nu} + \dot{\psi}_d - k_I e - c_1 S - c_2 \text{sign}(S)) \tag{6.34}$$

where $\dot{\psi}_d$ is from Eq. (6.28), $\hat{\alpha}$ and $\hat{\nu}$ are the estimation of α and ν, respectively, c_1 and c_2 are a positive constant, and

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0. \end{cases} \tag{6.35}$$

Rearrange Eq. (6.33) using $\psi_s = \psi_c - \psi$ as:

$$\begin{aligned} \dot{W}_1 &= S(\alpha\psi_s + \nu - \dot{\psi}_d + k_I e) \\ &= S\{(\alpha - \hat{\alpha})\psi_s + \hat{\alpha}\psi_s + \nu - \dot{\psi}_d + k_I e\} \\ &= S\{\tilde{\alpha}\psi_s + \hat{\alpha}(\psi_c - \psi) + \nu - \dot{\psi}_d + k_I e\} \end{aligned} \tag{6.36}$$

where $\tilde{\alpha} = \alpha - \hat{\alpha}$. Substituting ψ_c in Eq. (6.34) into Eq. (6.36) yields:

$$\begin{aligned} \dot{W}_1 &= S\{\tilde{\alpha}\psi_s - \hat{\nu} + \dot{\psi}_d - k_I e - c_1 S - c_2 \text{sign}(S) + \nu - \dot{\psi}_d + k_I e\} \\ &= S\{\tilde{\nu} - c_1 S - c_2 \text{sign}(S) + \tilde{\alpha}\psi_s\} \end{aligned} \tag{6.37}$$

where $\tilde{v} = v - \hat{v}$.

To obtain the adaptation rule for the parameter estimations of $\hat{v}$ and $\hat{\alpha}$, consider another Lyapunov function candidate W_2 as:

$$W_2 = \frac{1}{2}S^2 + \frac{1}{2}\gamma_v^{-1}\tilde{v}^2 + \frac{1}{2}\gamma_\alpha^{-1}\tilde{\alpha}^2 \tag{6.38}$$

where γ_v and γ_α are positive constants. Differentiating Eq. (6.38) and using Eq. (6.37) gives:

$$\begin{aligned} \dot{W}_2 &= S(\tilde{v} - c_1 S - c_2 \text{sign}(S) + \tilde{\alpha}\psi_s) + \gamma_v^{-1}\tilde{v}\dot{\tilde{v}} + \gamma_\alpha^{-1}\tilde{\alpha}\dot{\tilde{\alpha}} \\ &= -c_1 S^2 - c_2|S| + \tilde{v}(S - \gamma_v^{-1}\dot{\hat{v}}) + \tilde{\alpha}(\psi_s S - \gamma_\alpha^{-1}\dot{\hat{\alpha}}) \end{aligned} \tag{6.39}$$

where the relation $\dot{\tilde{v}} = -\dot{\hat{v}}$ and $\dot{\tilde{\alpha}} = -\dot{\hat{\alpha}}$ are used under the assumption that v and α are constant. Then, the adaptation law for $\hat{v}$ and $\hat{\alpha}$ can be obtained as:

$$\dot{\hat{v}} = \gamma_v S \tag{6.40}$$

$$\dot{\hat{\alpha}} = \gamma_\alpha \psi_s S. \tag{6.41}$$

Then,

$$\dot{W}_2 = -c_1 S^2 - c_2|S| \leq 0 \tag{6.42}$$

from which it can be concluded that S goes to zero in finite time, and finally the error state e tends to zero by LaSalle-Yoshizawa theorem [148]. This means that the heading angle ψ of the UAV can follow the desired heading ψ_d provided from Eq. (6.24) in spite of model uncertainties in α and v. To avoid the chattering problem which results from the discontinuity of sign function, this study replaces the sign function in the control command with the continuous saturation function as given:

$$\text{sat}\left(\frac{S}{\epsilon}\right) = \begin{cases} \frac{S}{\epsilon}, & \text{if } \left|\frac{S}{\epsilon}\right| \leq 1 \\ \text{sign}\left(\frac{S}{\epsilon}\right), & \text{otherwise} \end{cases} \tag{6.43}$$

where $\epsilon > 0$ represents the width of the boundary layer around the sliding surface.

6.2.2.2 Taking a target velocity into account

If the velocity of a ground moving target can be estimated (as will be described in the next section), the vector field can be adjusted in order to improve tracking guidance performance, by taking a target velocity into account. Let us consider the behaviour of a point orbiting a constant speed target at fixed radius r_d, then, the position of the point can be expressed as:

$$\begin{aligned} x_{tp} &= r_d \cos\theta + T_x t \\ y_{tp} &= r_d \sin\theta + T_y t \end{aligned} \tag{6.44}$$

where T_x and T_y are the speed of the target in x and y axis, respectively. By differentiating Eq. (6.44),

$$\begin{aligned} \dot{x}_{tp} &= -r_d \dot{\theta} \sin\theta + T_x \\ \dot{y}_{tp} &= r_d \dot{\theta} \cos\theta + T_y \end{aligned} \tag{6.45}$$

Then, the path heading angle for a moving target is obtained as:

$$\psi_p = \tan^{-1}\left(\frac{\dot{y}_{tp}}{\dot{x}_{tp}}\right) \tag{6.46}$$

Maintaining the desired UAV speed V_d leads the following condition to be met:

$$V_g^2 = (\dot{x}_{tp})^2 + (\dot{y}_{tp})^2 = (-r_d \dot{\theta} \sin\theta + T_x)^2 + (r_d \dot{\theta} \cos\theta + T_y)^2 \tag{6.47}$$

which simplifies to:

$$\dot{\theta}^2 (r_d^2) + \dot{\theta}(2 r_d T_y \cos\theta - 2 r_d T_x \sin\theta + (T_x^2 + T_y^2 - V_g^2) = 0 \tag{6.48}$$

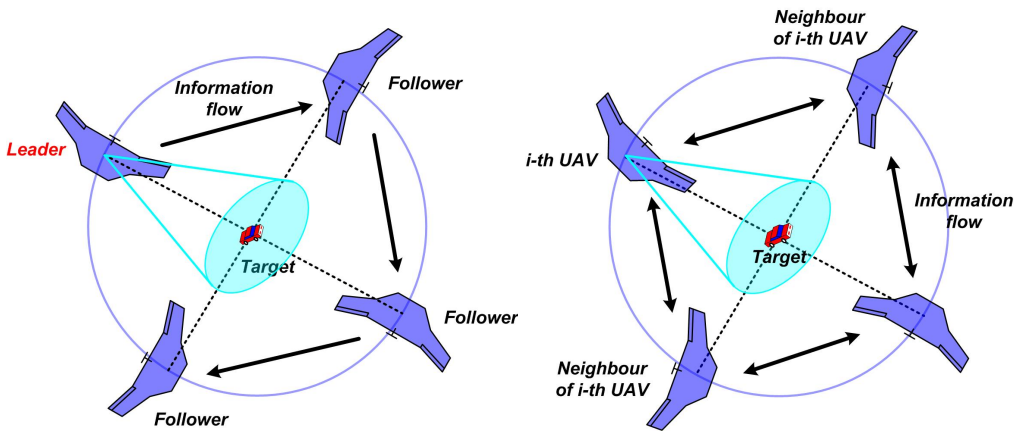
Substituting $\dot{\theta}$ obtained by solving Eq. (6.48) into Eq. (6.45) gives the final path heading angle ψ_p and the modified ψ_d from Eq. (6.24). Note that there will be an error due to the estimations of T_x and T_y which eventually could make the closed-loop dynamics unstable through the feedforward path. This can be addressed by acquiring as accurate estimations as possible (which will be discussed in Section 6.2.3.3) and selecting conservative c_1 and c_2 constants in the SMC controller in consideration of a bounded error for ψ_d.

6.2.3 Decentralised TVFG for Angular Separation

In performing a coordinated target tracking mission, UAVs should keep a prescribed inter-vehicle angular separation to maximise sensor coverage or enhance the estimation accuracy of the target information, while maintaining a standoff distance from a target. To do so, this section introduces decentralised angular separation control of multiple UAVs using either velocity or orbit radius change by different information architectures. It builds upon a rigid graph theory utilising: asymmetric minimally persistent leader-follower and symmetric nonminimally persistent, following the previous study [137].

6.2.3.1 Minimally persistent leader-follower information architecture

In this architecture, one of UAVs (leader) follows the standoff orbit around a target with desired airspeed and orbit radius using the TVFG with adaptive SMC. The remaining vehicles (followers) maintain the same orbit; they keep a prescribed angular spacing with the neighboring vehicle ahead of it by adjustments of airspeed or orbit radius. This architecture can be modelled by a directed graph as illustrated in Fig. 6.9(a), and requires a minimum possible number of communication/sensing links to achieve the circular orbit and angular spacing.



(a) Minimally persistent (b) Nonminimally persistent

Figure 6.9: Illustration of information architectures

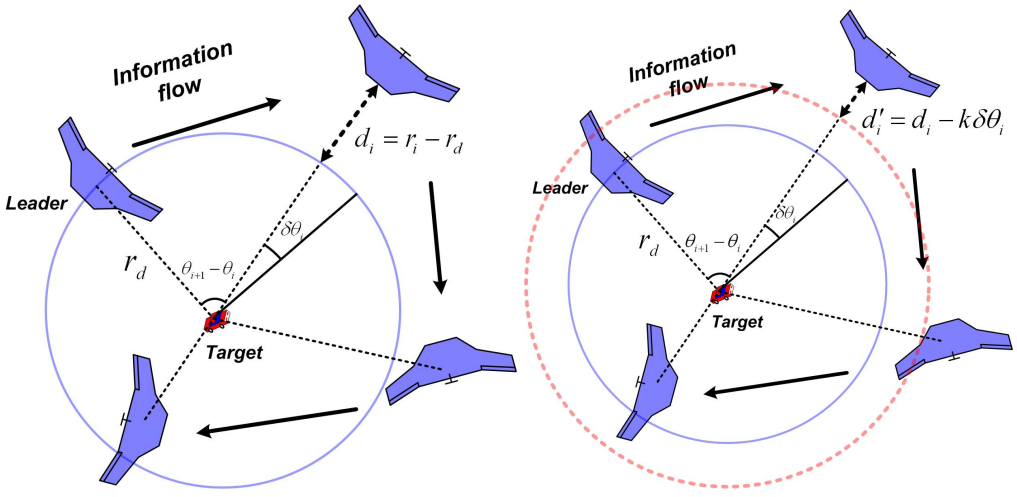
6.2.3.1.1 Orbit radius change Let n-th UAV have the constant desired orbit radius r_d, and the remaining $n-1$ UAVs have a variable orbit radius by changing the desired heading angle of the tangent vector field as:

$$\psi_i^d = \psi_i^p + \tan^{-1}(k_t(d_i - k_o^m \delta\theta_i^m)) \tag{6.49}$$

where k_o^m is a control gain weighting the convergence to a desired orbit, and $\delta\theta_i^m$ is i-th angular spacing error given by:

$$\delta\theta_i^m = \theta_{i+1} - \theta_i - \theta_d \tag{6.50}$$

where θ_d is the desired angular separation between UAVs. Note that, $d = r - r_d$ is a distance from the desired orbit to the UAV position in the normal TVFG as Eq. (6.24); now it is modified by additional term $(k_o^m \delta\theta_i^m)$ according to the angular spacing error, resulting in the temporary change of orbit radius, as illustrated in Fig. 6.10. In this figure, while the leader UAV is staying on the desired orbit, the follower UAV goes to the orbit of larger radius to decrease the angular separation error with the amount of $\delta\theta_i$ to the leader; the rest of followers will respond accordingly. The difference of time spent on the different orbits allows for the control of the angular separation of UAVs.



(a) Original orbit radius (b) Modified orbit radius

Figure 6.10: Illustration of orbit radius change in minimally persistent case

6.2.3.1.2 Velocity change Similarly to the orbit radius change case, let n-th UAV have the desired velocity input V_d, and the remaining $n-1$ UAVs have a

variable velocity input according to the angular spacing error as:

$$u_{vi} = V_d + \Delta V_{max} \tanh(k_v^m \delta\theta_i^m) \tag{6.51}$$

where k_v^m is a control gain, and $\delta\theta_i^m$ is the same as Eq. (6.50), and $\Delta V_{max} > 0$ is a design parameter to be met for a speed variation constraint of the UAV. When the angular separation is different from the desired value, a velocity input of each UAV is adjusted accordingly without changing an orbit radius, as illustrated in Fig. 6.11.

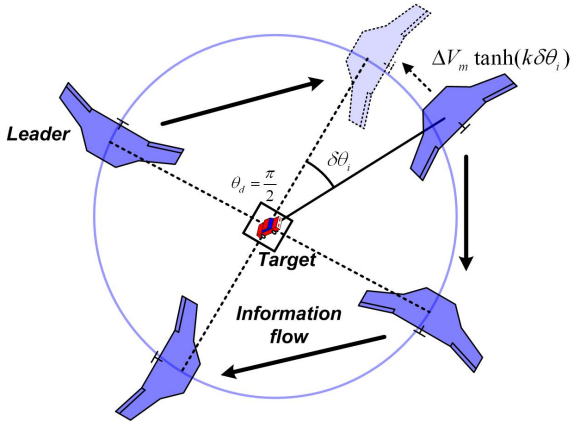


Figure 6.11: Illustration of velocity change in minimally persistent case

6.2.3.2 Nonminimally persistent information architecture

In this architecture, the airspeed or orbit radius is adjusted such that each vehicle moves toward the midpoint of its two nearest neighbours on the standoff orbit. This is modelled by an undirected graph as illustrated in Fig. 6.9(b). This architecture does not need to know the number of engaging vehicles in advance, compared to the minimally persistent case that requires a desired separation angle θ_d d; this control structure can be viewed as fully decentralised.

6.2.3.2.1 Orbit radius change Let i-th UAV have a variable orbit radius by changing a heading command of the TVFG as:

$$\psi_i^d = \psi_i^p + \tan^{-1}(k_t(d_i + k_o^n \delta\theta_i^n)) \tag{6.52}$$

where k_o^n is a control gain, and $\delta\theta_i^n$ is i-th angular spacing error given by:

$$\delta\theta_i^n = \theta_i - \frac{1}{2}(\theta_{i-1} + \theta_{i+1}) \tag{6.53}$$

where a modulo n cycle is formed around the circle as:

$$\theta_0 = \theta_n - 2\pi, \quad \theta_{n+1} = \theta_1 + 2\pi \tag{6.54}$$

6.2.3.2.2 Velocity change Let the n UAVs have a variable velocity input according to the angular spacing error as:

$$u_{vi} = V_d - \Delta V_{max} \tanh(k_v^n \delta\theta_i^n) \tag{6.55}$$

where k_v^n is a control gain, and $\delta\theta_i^n$ is the same as Eq. (6.53).

6.2.3.3 Decentralised target localisation

The performance of the standoff tracking guidance algorithms proposed so far is strongly coupled with the sensing and estimation capability against a moving target on the ground. To produce appropriate tracking data, a GMTI (Ground Moving Target Indicator) is a well-suited radar sensor due to its wide-coverage, all-weather, day/night, and real-time capabilities [84]. From this sensor data such as range, azimuth, or elevation of the target with respect to the sensor location, a certain level of accurate estimation could be obtained using conventional filtering techniques. In order to further improve the estimation accuracy, as well as cope with the proposed decentralised guidance approach, this study applies a decentralised multisensor fusion algorithm. Given that multiple UAVs carry out the process of tracking the same ground target, each UAV will obtain its own sensor measurement and executes the tracking filter separately. After each UAV receives the other's estimation via a communication link, it can run a decentralised data fusion algorithm. In this study, the decentralised extended Information filter (DEIF) and Kalman consensus algorithm are applied, as explained in Section 4.6.4.

6.2.4 Numerical Simulations

To verify the feasibility and benefits of the proposed approach, numerical simulations are performed by using realistic ground vehicle tracking scenario. The vehicle trajectory data acquired at 2 Hz in a S-Paramics [94] traffic model of Devizes, United Kingdom, are used to generate the GMTI measurements composed of relative range and azimuth angle with respect to a position of UAV, as shown

in Fig. 6.12 [96]. The ground vehicle departs at the western side of Devizes and traverses a part of the town center and then turns back until the journey ends at the northwestern side of Devizes. These simulated GMTI measurements of UAVs were mixed with the white noise having the following standard deviations:

$$\begin{aligned} \text{UAV1:} \quad & (\sigma_{r1}, \sigma_{\phi 1}) = (10\ m, 6\ deg) \\ \text{UAV2:} \quad & (\sigma_{r2}, \sigma_{\phi 2}) = (13\ m, 5\ deg) \\ \text{UAV3:} \quad & (\sigma_{r3}, \sigma_{\phi 3}) = (15\ m, 4\ deg) \end{aligned}$$

Figure 6.13 shows the ground target estimation result using the EIF for each UAV. As can be seen in this figure, the ground vehicle moves complicatedly while frequently changing the speed and the direction of driving. The other parameters used in the simulation are shown in Table 6.1.

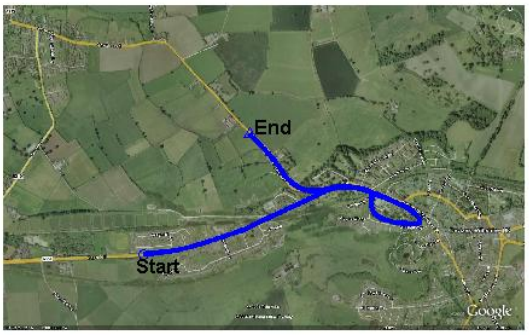
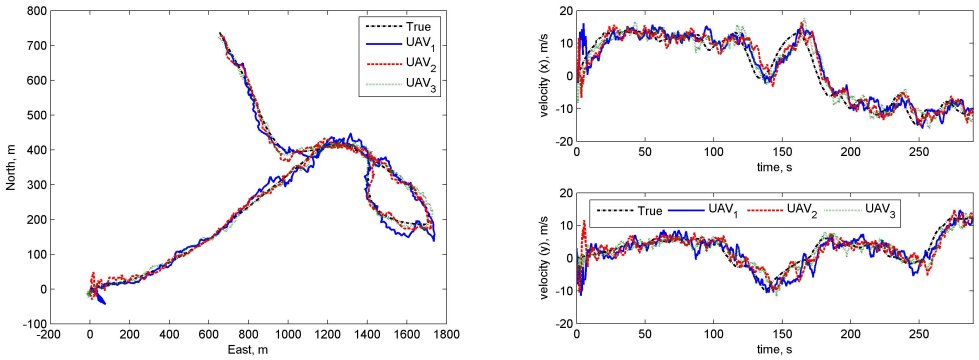


Figure 6.12: The Scenario description in the civilian traffic at Devizes, Wiltshire, UK

First, ground target tracking is performed using two UAVs without coordination (i.e. no angular separation control) to investigate the performance of the TVFG using adaptive SMC. Figure 6.14 shows trajectories of UAVs relative to a target using with and without estimated target velocity. As can be seen in this figure, incorporating velocity information into the TVFG command greatly improves tracking guidance performance. Figure 6.15 displays the relative trajectories of UAVs with respect to the ground vehicle, and Fig. 6.16 shows relative distance histories between a target and UAVs, in case that there are uncertainty and disturbances in the heading-hold loop. Table 6.2 indicates that the standoff

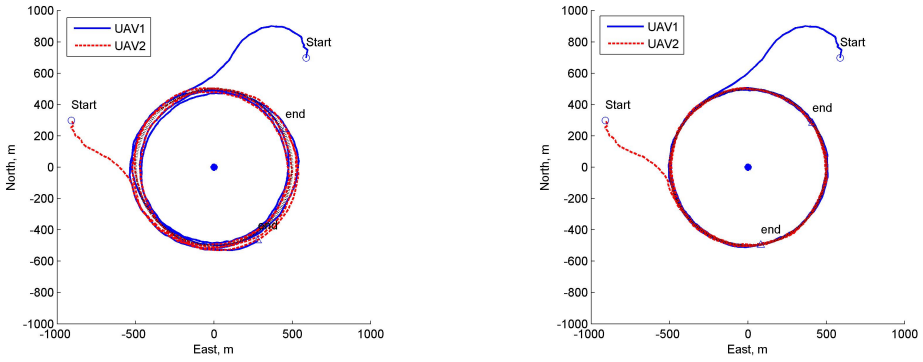


(a) Trajectory (b) Velocity

Figure 6.13: Ground target estimation results using the EIF

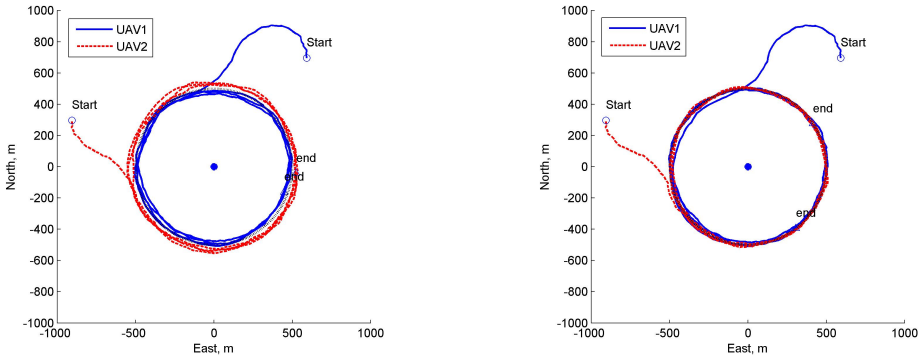
Table 6.1: Simulation parameters

Parameters	Value
Standoff radius r_d	500 m
Angular separation θ_d	$\frac{2}{3}\pi$
Desired UAV speed V_g	40 m/s
Speed increment ΔV_{max}	12 m/s
Time delay constant τ_v, τ_ω	$\frac{1}{3}$ sec
Gain [k_v, k_o, k_I, k_t, c_1, c_2]	[30, 3, 0.02, 0.015, 0.3, 0.6]
Maximum turning rate ω_{max}	0.2 rad/s
Heading time constant [α, $\hat{\alpha}_0$]	[1.5, 3.0]
Disturbance [ν, $\hat{\nu}_0$]	[(0.2,-0.2,0.15), 0.0]



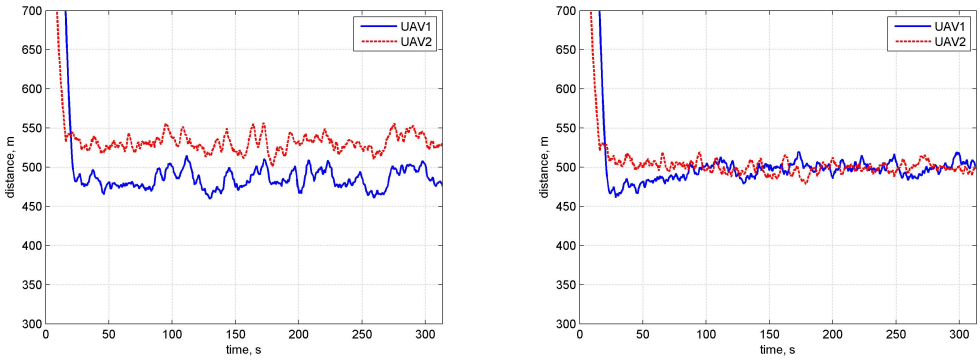
(a) Without estimated target velocity (b) With estimated target velocity

Figure 6.14: Trajectories of UAVs relative to a moving target using SMC



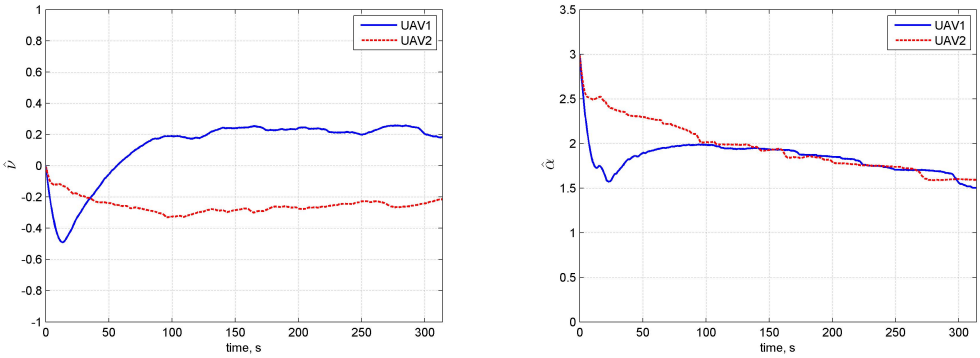
(a) SMC *(b) Adaptive SMC*

Figure 6.15: Trajectories of UAVs relative to a moving target with uncertainty and disturbance in a heading-hold loop



(a) SMC *(b) Adaptive SMC*

Figure 6.16: Distance histories between a moving target and UAVs



(a) Disturbance $\hat{\nu}$ *(b) Time constant $\hat{\alpha}$*

Figure 6.17: Estimation of unknown parameters in adaptive SMC

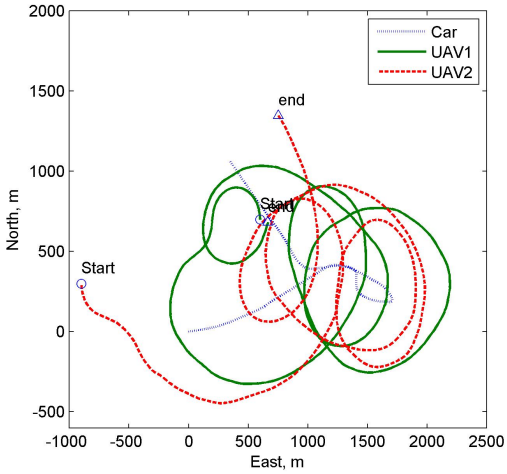


Figure 6.18: Final trajectory of a moving target and UAVs

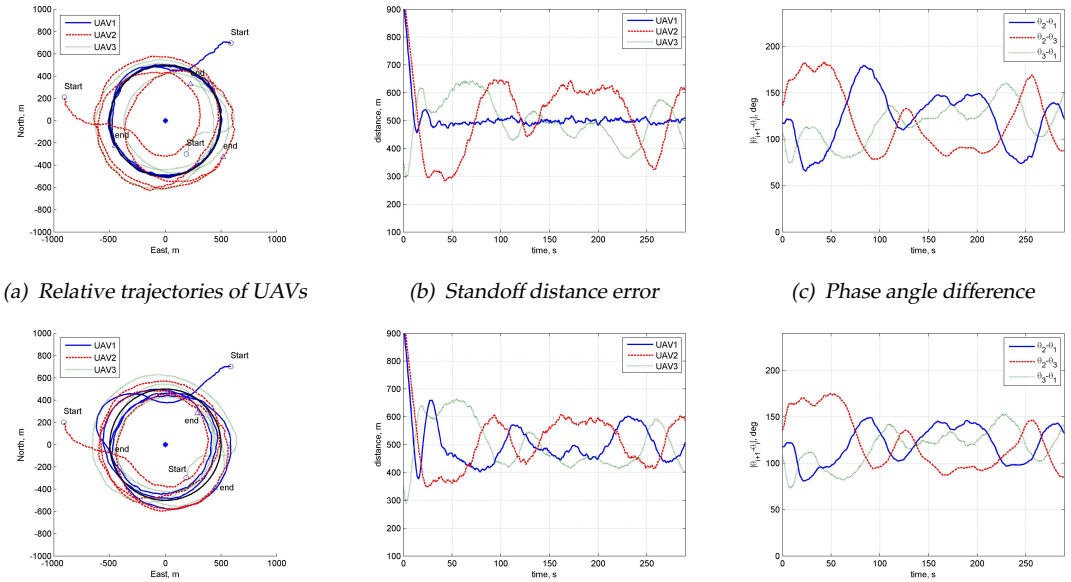
Table 6.2: Performance comparison of guidance algorithms with different information/communication structures

Mean error	Single UAV (EIF)		Multiple UAVs (DEIF)			
	SMC	ASMC	Orbit radius change		Velocity change	
			Minimal	Nonminimal	Minimal	Nonminimal
Position (m)	14.3291		7.6990	4.9703	7.6790	4.7218
Velocity (m/s)	3.1445		2.5216	2.1412	2.5082	2.0913
Standoff distance (m)	26.8211	14.4717	63.3150	70.5503	15.9357	14.0693
Angular separation (deg)	-	-	23.9987	16.7073	11.6899	2.7884

tracking performance of adaptive SMC is much better than that of normal SMC due to the estimation of unknown parameters as shown in Fig. 6.17. Figure 6.18 shows the final absolute trajectory of a target and UAVs using the TVFG using adaptive SMC.

In addition, Table 6.2 shows mean errors in the decentralised estimation and the decentralised TVFG using adaptive SMC with different information/communication structures for three UAVs. In this table, 'Minimal' and 'Nonminimal' columns represent the use of a minimum communication network (i.e. only $g_k^{21}, g_k^{32}, g_k^{13} = 1$) with minimally persistent information architecture and a fully-connected communication link with nonmnimally persistent one, respectively. In this table and Fig. 6.20, the velocity change scheme with a fully-connected communication network and nonminimally information architecture shows the best performance in terms of the estimation accuracy, standoff distance, and phase keeping among

UAVs. However, the minimum communication/information structure requires less communication between UAVs and velocity control efforts as shown in Table 6.2 and Figs. 6.19~6.20. This can be a preferable option with its reasonable guidance performance depnding on the mission specification. Moreover, although the velocity change scheme shows much better performance than that of orbit radius change, in case that frequent velocity change is undesirable or unattainable, the angular separation can be achieved without velocity control but with a bounded error. Note that this is done by adjusting orbit radius appropriately at the expense of performance in standoff distance as shown in Fig. 6.19.

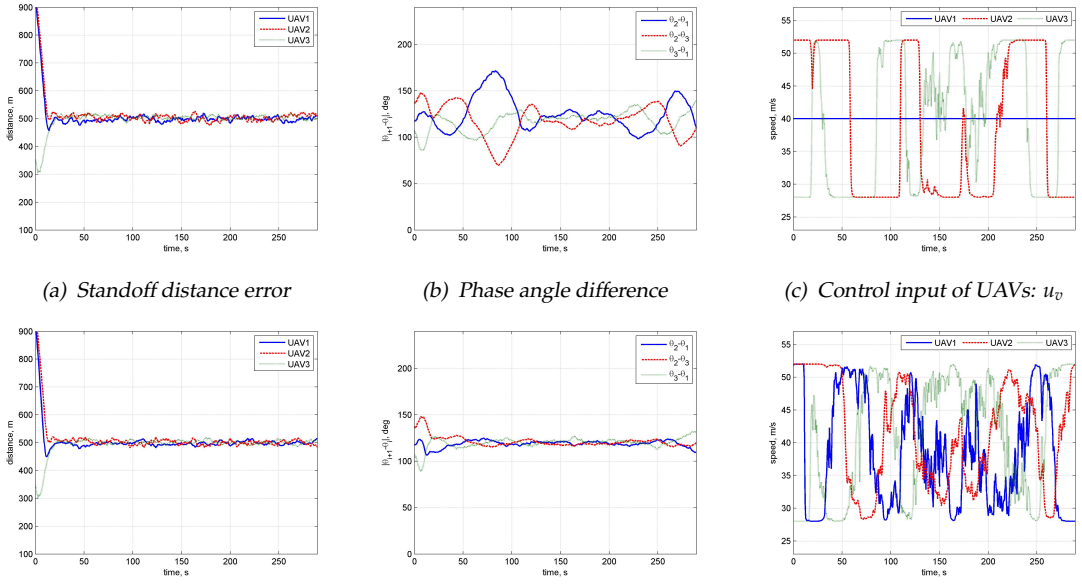


(a) Relative trajectories of UAVs (b) Standoff distance error (c) Phase angle difference

(d) Relative trajectories of UAVs (e) Standoff distance error (f) Phase angle difference

Figure 6.19: Tracking results using orbit radius change for angular separation and DEIF: (a)~(c) minimally persistent persistent information architecture & minimum communication link; (d)~(f) nonminimally persistent & fully-connected communication link

Table 6.3 compares the performance in the decentralised estimation based on the Kalman consensus algorithm and guidance with communication noises and networks. This result is acquired from the simulation using the EIF for each UAV operating at 2 Hz along with sharing/averaging information using the Kalman consensus algorithm at 8 Hz. If there is no communication noise, using the fully-connected network shows better performance again than that of the minimum network. Meanwhile, in case that there are noises which can be caused by packet



(a) Standoff distance error (b) Phase angle difference (c) Control input of UAVs: u_v

(d) Standoff distance error (e) Phase angle difference (f) Control input of UAVs: u_v

Figure 6.20: Tracking results using velocity change for angular separation and DEIF: (a)~(c) minimally persistent persistent information architecture & minimum communication link; (d)~(f) nonminimally persistent & fully-connected communication link

loss or irregular delay, the fully-connected one could accumulate the noise and thus degrade the estimation accuracy as found in Table 6.3. Moreover, it is shown that by tuning system error covariance Q_k^c and the noise covariance Ω_k, representing how much the estimates rely on system dynamics and information from the other UAVs via communications, the estimation accuracy can be enhanced.

Table 6.3: Performance comparison of the Kalman consensus algorithm with different communication noises/networks

Mean error	Without noise		$Q_k^c = 0.1, \Omega_k = 0.02$		$Q_k^c = 0.001, \Omega_k = 0.02$	
	Minimal	Nonminimal	Minimal	Nonminimal	Minimal	Nonminimal
Position (m)	5.4660	5.3028	5.8618	6.8784	5.7337	6.2565
Velocity (m/s)	2.3406	2.3339	2.7461	3.6148	2.6570	3.1985
Standoff distance (m)	15.7559	14.3813	16.0396	15.3933	15.9617	14.9471
Phase keeping (deg)	11.7294	2.7728	11.7436	2.8121	11.7440	2.8150

6.3 Rendezvous and Sandoff Tracking Using Differential Geometry

This section proposes UAV rendezvous and standoff tracking guidance to a moving ground target using differential geometry motivated by previous works [149, 150]. Using the relative geometry, convergent, divergent and parallel solutions can be obtained depending on their initial positions and the velocity ratio between them. Then, the convergent solution can be used to guide the UAV on the corresponding rendezvous geometry associated with the target movement. In a similar way, a novel guidance law for standoff tracking is derived by superimposing a standoff orbit circle around the target position. In order to verify the feasibility and benefits of the proposed approach, numerical simulations are performed using a realistic scenario, in which two cooperative UAVs are equipped with GMTI (Ground Moving Target Indicator) sensor and run the EKF with state-vector fusion for target localisation.

The proposed differential geometric guidance law has several advantages along with its inherent simplicity over the other standoff tracking guidance laws. First of all, whilst most of the literature have focused on the stability analysis limited to standoff tracking of a fixed target, the proposed approach can analyse stability for both rendezvous and standoff tracking of a moving target. Another benefit is that the guidance command can explicitly consider a target velocity for enhancing the tracking performance when its estimation by the localisation filter is reasonably accurate. Lastly, the proposed guidance law requires the reduced number of tuning variables, only a curvature command, unlike other approaches such as vector field guidance requiring more parameters for appropriate vector field generation as well as guiding vehicle into the field [139, 151].

The remainder of this section is organised as follows. Section 6.3.1 introduces rendezvous and standoff tracking geometry between the UAV and the target using differential geometry. Section 6.3.2 proposes the guidance algorithms to generate the UAV turn-rate command for both rendezvous and standoff tracking against a moving target and then proves its global convergence using Lyapunov theory. Lastly, numerical simulations are performed to verify the performances of the proposed methodology in Section 6.3.3.

6.3.1 Differential Geometry Associated With UAV and Target

6.3.1.1 Rendezvous geometry

Consider an UAV and a moving target with their motion associated each other in a Frenet-Serret frame [149, 152]. Rendezvous geometry is built up using the two-dimensional Frenet-Serret frame defined by a tangent vector **t** and a normal vector **n** of each vehicle as shown in Fig. 6.35. The required tangent direction of

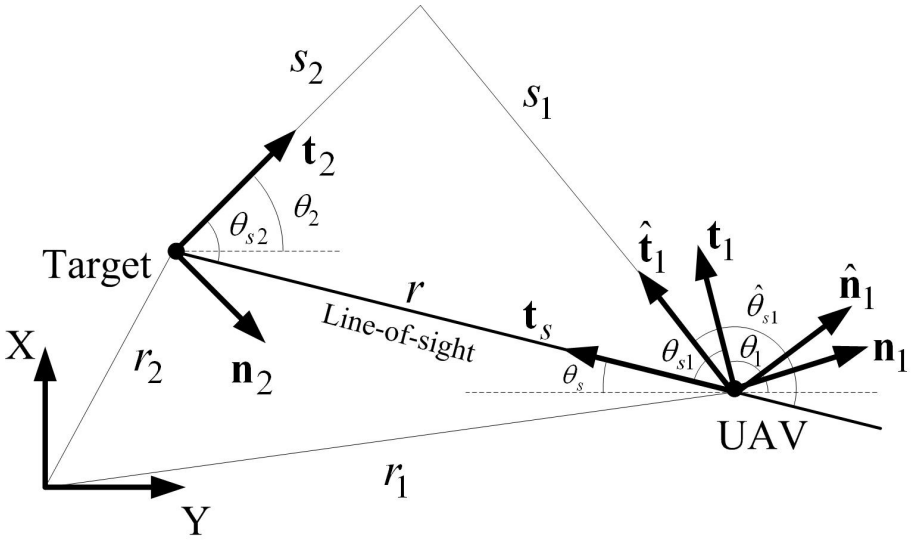


Figure 6.21: Guidance geometry

the UAV for rendezvous, $\hat{\mathbf{t}}_1$, is represented with the following vector addition on the rendezvous triangle in Fig. 6.35:

$$s_1\hat{\mathbf{t}}_1 = r\mathbf{t}_s + s_2\mathbf{t}_2 \tag{6.56}$$

where $\mathbf{t}_2$ and $\mathbf{t}_s$ are a current velocity vector of the target and a line-of-sight (LOS) vector from the UAV to the target, respectively, r is a distance between the UAV and the target, and s_1 and s_2 are the resultant lengths of the tangent vectors to the rendezvous point. Let us define a velocity ratio of the UAV and the target, γ, as:

$$\gamma = \frac{v_1}{v_2} = \frac{s_1}{s_2} \tag{6.57}$$

where v_i is the constant speed of each vehicle. Then, Eq. (6.56) changes to:

$$\hat{\mathbf{t}}_1 = \frac{1}{\gamma}\left[\frac{r}{s_2}\mathbf{t}_s + \mathbf{t}_2\right]. \tag{6.58}$$

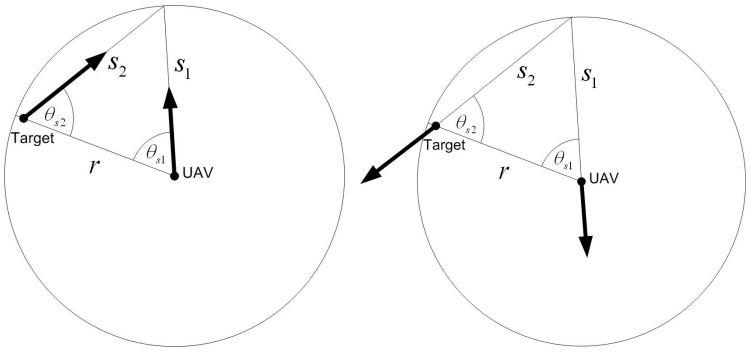
Applying a cosine rule to the geometry around the LOSs to the target gives:

$$\left(\frac{r}{s_2}\right)^2 - 2\cos(\theta_{s2})\left(\frac{r}{s_2}\right) - \left(\gamma^2 - 1\right) = 0. \tag{6.59}$$

Solving this equation gives:

$$\left(\frac{r}{s_2}\right) = \cos(\theta_{s2}) \pm \sqrt{\gamma^2 - \sin^2(\theta_{s2})}. \tag{6.60}$$

For this case of a straight line rendezvous, the triangle of the rendezvous point and vehicle positions is thus invariant in shape, but will shrink or grow as UAVs travel along the solution. One solution will result in rendezvous, the other in a divergent geometry as shown in Figure 6.22. In this figure, a circle with radius of



(a) Convergent Solution *(b) Divergent Solution*

Figure 6.22: Solution geometries for rendezvous

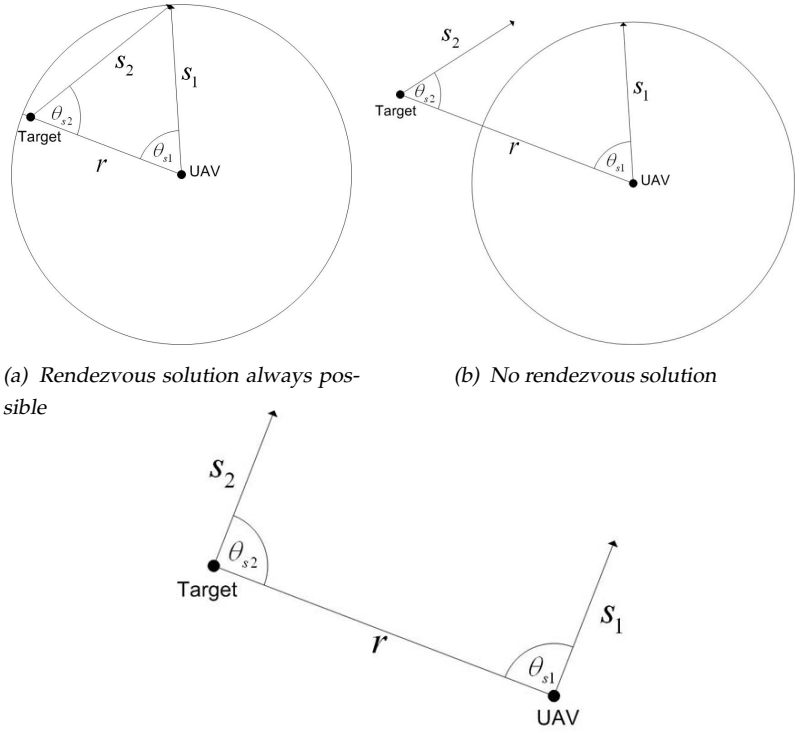
s_1 is drawn to visualise the reachable area of the UAV considering the length (or velocity) ratio $\gamma = \frac{s_1}{s_2}$ as in Eq. (6.57), and possible initial conditions for rendezvous geometry between the UAV and target. The convergent solution uses the positive square root, whilst the divergent solution uses the negative square root. Hence for this case, the solution will always use the positive square root, as:

$$\left(\frac{r}{s_2}\right) = \cos(\theta_{s2}) + \sqrt{\gamma^2 - \sin^2(\theta_{s2})}. \tag{6.61}$$

It is also worth exploring the solution as a function of the speed ratio γ. If $\gamma > 1$, then the expression within the square root is always greater than zero for all θ_{s2}, or:

$$\gamma^2 - \sin^2(\theta_{s2}) > 0, \quad \gamma > 1. \tag{6.62}$$

This means that there will always be a real solution for any initial geometry. Such a condition is shown in Fig. 6.23(a). However, if $\gamma < 1$, then there will be some θ_{s2}



(a) Rendezvous solution always possible

(b) No rendezvous solution

(c) Parallel solution

Figure 6.23: Rendezvous solutions depending on initial positions and γ

for which:

$$\gamma^2 - \sin^2(\theta_{s2}) < 0, \quad \gamma < 1. \tag{6.63}$$

This implies that for some geometry, a real solution is not possible, and the UAV will not be able to rendezvous with the target. For example, for the fixed target direction shown in Fig. 6.23(b), the circle indicating all the possible travel directions of the UAV has no intersection with the trajectory of the target abiding by the length ratio γ, and thus there is no solution. For the case of the two vehicles having the same speed, i.e. $\gamma = 1$, the rendezvous geometry equation now becomes:

$$\hat{\mathbf{t}}_1 = \left[\frac{r}{s_2}\mathbf{t}_s + \mathbf{t}_2\right] \tag{6.64}$$

with

$$\left(\frac{r}{s_2}\right)^2 - 2\cos(\theta_{s2})\left(\frac{r}{s_2}\right) = 0. \tag{6.65}$$

The solution for this case is given by:

$$\left(\frac{r}{s_2}\right) = 2\cos(\theta_{s2}) \text{ or } 0. \tag{6.66}$$

There is no longer an imaginary solution, and now two real solutions are given. The first solution implies a geometry which gives rise to an isosceles triangle solution as before, but only where $s_1 = s_2$. Since the solution depends on the initial positions and orientation of the target, global convergence is not possible for this case. The second solution implies that the ratio r/s_2 is zero. For $r > 0$, this implies $s_2 = \infty$ or $\theta_{s1} = \theta_{s2} = \pi/2$. This solution results in the UAV and the target moving on parallel courses that neither converge or diverge as shown in Fig. 6.23(c), and the vehicles will maintain the geometry both in shape and size for this solution. Hence this solution is of interest when the vehicles are required to move in some form of group or formation where they will retain some sort of cohesion in manoeuvre. Note that this condition does not imply any particular geometry, much as the other solutions do. All that is required is that the ratio of the two paths s_1 and s_2 in the solution are the same as the speed ratio between them.

The guidance algorithm must therefore compute the required velocity vector tangent $\hat{\mathbf{t}}_1$ using Eq. (6.58) along with Eq. (6.60) for the UAV and produce a closed loop system which drives the current tangent vector $\mathbf{t}_1$ onto it in a stable manner.

6.3.1.2 Standoff tracking geometry

This section extends the rendezvous problem of the previous section into standoff tracking for which the UAV needs to track the moving target while maintaining a certain distance from it. In a similar way to the previous section, let us consider the associated geometry of the UAV and the target with newly adopting a standoff distance d_m. The relative velocity of the UAV with respect to the target is shown in Fig. 6.24 for rendezvous. The relative velocity of the UAV with respect to the target is given by:

$$\mathbf{v}_r = \mathbf{v}_1 - \mathbf{v}_2 \tag{6.67}$$

The condition for rendezvous is shown to be such that the relative velocity vector should lie along the LOS between the UAV and the target. This ensures that

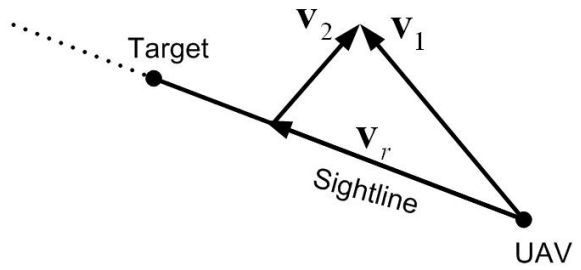


Figure 6.24: Relative velocity for rendezvous

the geometry does not change over time as the rendezvous triangle shrinks but maintains its shape. This is consistent with the fact that the solution of r/s is constant as defined in Eq. (6.60). This can now be modified to produce the geometry of standoff tracking by superimposing a circle of radius d_m around the target as shown in Fig. 6.25. If the relative velocity vector $\mathbf{v}_r$ is aligned with the

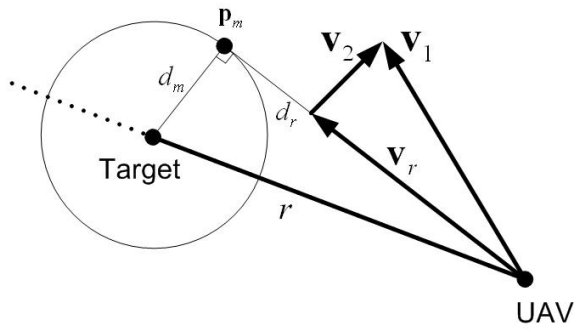


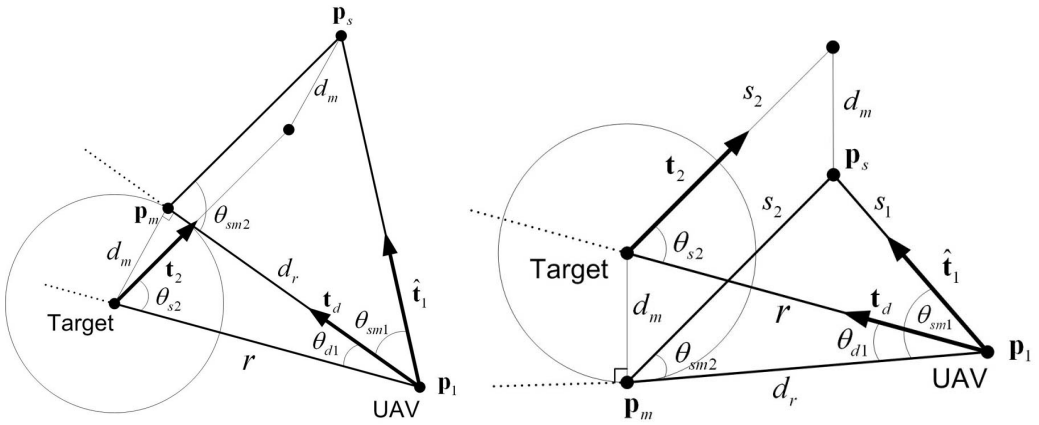
Figure 6.25: Relative velocity for standoff tracking orbit approach

tangent line to the standoff circle, then the rendezvous point for standoff tracking will be at $\mathbf{p_m}$.

Now, the rendezvous geometry shown in Fig. 6.35 is modified to fit the standoff tracking as shown in Fig. 6.26. Herein, the original rendezvous triangle is modified into the triangle given by $\{\mathbf{p_1}, \mathbf{p_m}, \mathbf{p_s}\}$, for both a clockwise and anti-clockwise rotation from the LOS to $\mathbf{p_m}$. The vector sum on the engagement geometry is given as:

$$\hat{\mathbf{t}}_1 = \frac{1}{\gamma}\left[\frac{d_r}{s_2}\mathbf{t}_d + \mathbf{t}_2\right] \tag{6.68}$$

where $\mathbf{t}_d$ is the unit tangent vector from the UAV to $\mathbf{p_m}$, and d_r is the distance between the UAV and rendezvous point $\mathbf{p_m}$. Applying the cosine rule to this



(a) Clockwise solution (b) Anti-clockwise solution

Figure 6.26: Geometry for standoff tracking

geometry gives:

$$\left(\frac{d_r}{s_2}\right)^2 - 2\cos(\theta_{sm2})\left(\frac{d_r}{s_2}\right) - (\gamma^2 - 1) = 0 \tag{6.69}$$

where

$$d_r = \sqrt{r^2 - d_m^2} \tag{6.70}$$

$$\theta_{sm2} = \theta_{s2} \pm \theta_{d1}. \tag{6.71}$$

Herein, θ_{d1} is either added for a clockwise solution and subtracted for an anti-clockwise solution. Substituting the following relations:

$$\cos(\theta_{d1}) = \frac{d_r}{r} = \frac{\sqrt{r^2 - d_m^2}}{r} \tag{6.72}$$

$$\sin(\theta_{d1}) = \frac{d_m}{r} \tag{6.73}$$

into

$$\cos(\theta_{sm2}) = \cos(\theta_{s2} \pm \theta_{d1}) = \cos(\theta_{s2})\cos(\theta_{d1}) \mp \sin(\theta_{s2})\sin(\theta_{d1}) \tag{6.74}$$

gives

$$\cos(\theta_{sm2}) = \frac{\sqrt{r^2 - d_m^2}}{r}\cos(\theta_{s2}) \mp \frac{d_m}{r}\sin(\theta_{s2}). \tag{6.75}$$

Note that as the standoff distance is reached, the parameters become:

$$r \to d_m,\ d_r \to 0,\ \theta_{d1} \to \frac{\pi}{2}.$$

Although the geometry of this case is not fixed with respect to the LOS, the solution requires the relative velocity vector to lie along the tangent line vector $\mathbf{t}_d$ which will not be changed, and the two vehicle velocity vectors are fixed. Therefore, the triangle $\{\mathbf{p_1}, \mathbf{p_m}, \mathbf{p_s}\}$ is fixed in shape and orientation, and will shrink as the UAV approaches to the standoff orbit $\mathbf{p_m}$. Then, the ratio d_r/s_2, a solution to Eq. (6.69), will have a fixed solution as:

$$\left(\frac{d_r}{s_2}\right) = \cos(\theta_{sm2}) \pm \sqrt{\gamma^2 - \sin^2(\theta_{sm2})}. \tag{6.76}$$

The guidance algorithm must therefore compute the required velocity vector $\hat{\mathbf{t}}_1$ using Eq. (6.68) along with Eq. (6.76) for the UAV and produce a closed loop system which drives the current tangent vector $\mathbf{t}_1$ onto it in a stable manner, as in the case for rendezvous.

6.3.2 UAV Guidance Law for Rendezvous and Standoff Tracking

This section proposes the guidance laws for both rendezvous and standoff tracking case. For ease of deriving the guidance laws, this section assumes that at the current sampling time, the target is instantaneously non-manoeuvring but its velocity is exploited, which can be estimated by tracking filters of the UAVs.

6.3.2.1 Rendezvous case

A guidance law is designed for rendezvous covering all the convergent solutions discussed in the previous section. Assuming the UAV has a speed advantage over the target, the positive solutions are considered from Eq. (6.59). Let us define the angle between the required tangent vector $\hat{\mathbf{t}}_1$ and the current UAV tangent vector $\mathbf{t}_1$ as:

$$\chi_1 = (\pi - \theta_{s1}) - \hat{\theta}_{s1} \tag{6.77}$$

where $(\pi - \theta_{s1})$ and $\hat{\theta}_{s1}$ are the current UAV tangent angle and the required tangent angle with respect to the LOS, respectively, as shown in Fig. 6.35. To guide the UAV onto the required geometry, considering an actively rotating tangent vector

$\mathbf{t}_1$, the guidance command u_ω for turn rate is set by using a curvature command as:

$$u_\omega = \dot{\theta}_1 = \kappa_1 v_1 \tag{6.78}$$

where v_1 is the UAV speed, and κ_1 is the curvature command. For ease of analysis, the turn rate $\dot{\theta}_1$ will be used rather than the explicit κ command.

To develop the guidance algorithm and assess the resulting stability, consider a Lyapunov function as:

$$L_1 = \frac{1}{2}\chi_1^2. \tag{6.79}$$

Its time derivative is given by:

$$\dot{L}_1 = \chi_1 \dot{\chi}_1 \tag{6.80}$$

where

$$\dot{\chi}_1 = -\dot{\theta}_{s1} - \dot{\hat{\theta}}_{s1}. \tag{6.81}$$

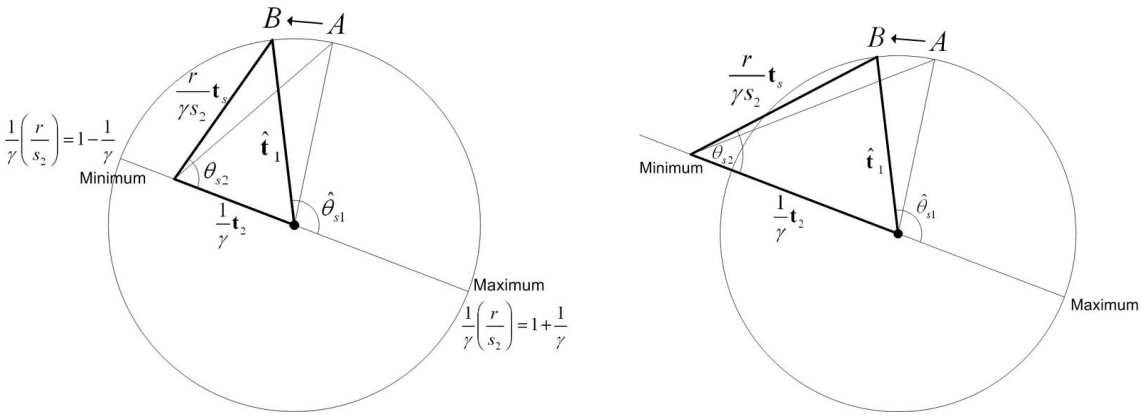
The differential of the required angle $\hat{\theta}_{s1}$ is obtained from examining the rate of change of $\hat{\mathbf{t}}_1$ as:

$$\hat{\mathbf{t}}_1 = \frac{1}{\gamma}\left[\left(\frac{r}{s_2}\right)\mathbf{t}_s + \mathbf{t}_2\right] \tag{6.82}$$

Differentiating this gives:

$$\dot{\hat{\mathbf{t}}}_1 = \frac{1}{\gamma}\left[\frac{d}{dt}\left(\frac{r}{s_2}\right)\mathbf{t}_s + \left(\frac{r}{s_2}\right)\dot{\mathbf{t}}_s\right] = \dot{\hat{\theta}}_1 \hat{\mathbf{n}}_1 \tag{6.83}$$

where $\hat{\mathbf{n}}_1$ is the normal vector to $\hat{\mathbf{t}}_1$, and $\dot{\mathbf{t}}_2 = 0$ is used from the assumption a target is instantaneously non-manoeuvring at the current sampling time. A geometric interpretation of Eq. (6.82) is reproduced in Fig. 6.27 to make use of the relation between corresponding angles (especially for θ_{s2} and $\hat{\theta}_{s1}$) intuitively. As the value of r/s_2 exists between its maximum ($1+\gamma$ at $\theta_{s2} = 0$) and minimum ($1-\gamma$ at $\theta_{s2} = \pi$) bounds from Eq. (6.61), the magnitude of vector $\mathbf{t}_s$, $r/(\gamma s_2)$, has its maximum and minimum values. Figure 6.27 shows that as the engagement geometry changes by the guidance, the solution $\hat{\mathbf{t}}_1$ will change and rotate around the circle, and the rotation of the solution vector $\hat{\mathbf{t}}_1$ is related to the rotation of the LOS vector



(a) Geometry $\gamma > 1$ (b) Geometry $\gamma < 1$

Figure 6.27: Geometric interpretation of rendezvous solutions

$\mathbf{t}_s$. As shown in Fig. 6.27, as solution A moves to solution B when $\gamma > 1$, $\hat{\theta}_{s1}$ as well as the target to LOS angle θ_{s2} increases, and hence there is a monotonic relationship between these two angles. Meanwhile, for the $\gamma < 1$ case, the angles oscillates between maximum and minimum bounds according to the condition from Eq. (6.61). Hence, the following equation holds:

$$\dot{\hat{\theta}}_{s1} = \lambda_1(r/s_2, \theta_{s2})\dot{\theta}_{s2} \tag{6.84}$$

From Fig. 6.35 and Fig. 6.27, the relations between the angles are given as:

$$\dot{\theta}_{s2} = \dot{\theta}_s \tag{6.85}$$

$$\dot{\hat{\theta}}_{s1} = \dot{\hat{\theta}}_1 + \dot{\theta}_s \tag{6.86}$$

since

$$\theta_{s2} = \theta_2 + \theta_s \tag{6.87}$$

$$\hat{\theta}_{s1} = \theta_s + \hat{\theta}_1 \tag{6.88}$$

and $\dot{\theta}_2 = 0$. Combining Eq. (6.84) and Eqs. (6.85)~(6.86) gives:

$$\dot{\hat{\theta}}_{s1} = \dot{\hat{\theta}}_1 + \dot{\theta}_s = \lambda_1(r/s_2, \theta_{s2})\dot{\theta}_s. \tag{6.89}$$

Rephrasing this yields:

$$\dot{\hat{\theta}}_1 = -\left[1 - \lambda_1(r/s_2, \theta_{s2})\right]\dot{\theta}_s. \tag{6.90}$$

The time derivative of Lyapunov candidate function in Eq. (6.81) can be rephrased by using Eq. (6.90) as:

$$\dot{L}_1 = \chi_1 \left[\dot{\theta}_1 + (1 - \lambda_1)\, \dot{\theta}_s \right]. \tag{6.91}$$

since

$$\dot{\chi}_1 = -\dot{\theta}_{s1} - \dot{\hat{\theta}}_{s1} = \dot{\theta}_1 + \dot{\theta}_s - (\dot{\hat{\theta}}_1 + \dot{\theta}_s) = \dot{\theta}_1 - \dot{\hat{\theta}}_1 \tag{6.92}$$

where

$$\theta_{s1} = (\pi - \theta_1) - \theta_s \tag{6.93}$$

To find the bounds of $1 - \lambda_1$, first of all, the differential of r/s_2 is obtained by differentiating Eq. (6.59) as:

$$\frac{d}{dt}\left(\frac{r}{s_2}\right) = \frac{-\sin(\theta_{s2})\left(\frac{r}{s_2}\right)}{\left(\frac{r}{s_2}\right) - \cos(\theta_{s2})}\dot{\theta}_s. \tag{6.94}$$

Substituting Eq. (6.94) and Eq. (6.90) into Eq. (6.83) gives:

$$\begin{aligned} &\frac{1}{\gamma}\left[\left(\frac{-\sin(\theta_{s2})\left(\frac{r}{s_2}\right)}{\left(\frac{r}{s_2}\right) - \cos(\theta_{s2})}\right)\mathbf{t}_s + \left(\frac{r}{s_2}\right)\mathbf{n}_s\right]\dot{\theta}_s \\ &\qquad = -\left[1 - \lambda_1(r/s_2, \theta_{s2})\right]\dot{\theta}_s \hat{\mathbf{n}}_1 \end{aligned} \tag{6.95}$$

Taking a norm of both sides of this equation gives:

$$|1 - \lambda_1| = \frac{\frac{1}{\gamma}\left(\frac{r}{s_2}\right)\sqrt{\left(\frac{r}{s_2}\right)^2 - 2\cos(\theta_{s2})\left(\frac{r}{s_2}\right) + 1}}{\left|\left(\frac{r}{s_2}\right) - \cos(\theta_{s2})\right|}. \tag{6.96}$$

Using Eq. (6.59) for the term under the square root of the above equation gives:

$$|1 - \lambda_1| = \frac{\left(\frac{r}{s_2}\right)}{\left|\left(\frac{r}{s_2}\right) - \cos(\theta_{s2})\right|}. \tag{6.97}$$

Combining above equation and Eq. (6.61) gives:

$$-\left(1 + \frac{1}{\gamma}\right) \leq (1 - \lambda_1(r/s_2, \theta_{s2})) \leq \left(1 + \frac{1}{\gamma}\right) \tag{6.98}$$

Then, the guidance command to UAV, turn rate $\dot{\theta}_1$, can be set as:

$$\dot{\theta}_1 = -\left(1 + \frac{1}{\gamma}\right)\left|\dot{\theta}_s\right| \text{sign}(\chi_1) - k_\chi \chi_1 \tag{6.99}$$

where control gain $k_\chi > 0$. The resulting Lyapunov rate is now:

$$\dot{L}_1 = \chi_1[-\left(1 + \frac{1}{\gamma}\right)\left|\dot{\theta}_s\right| \text{sign}(\chi_1) + (1 - \lambda_1)\,\dot{\theta}_s] - k_\chi \chi_1^2 \leq 0. \tag{6.100}$$

$\dot{L}_1$ is negative semi-definite since Eq. (6.98) makes an absolute magnitude of the first term in the square brackets equal to or greater than that of the second term. Thus the guidance law derived herein will produce a stable convergence of the UAV onto a rendezvous with the target.

6.3.2.2 Standoff target tracking case

As the modified geometry for the standoff tracking problem was shown to be also invariant, its guidance law can be obtained in a similar way to the rendezvous case. Lyapunov function is identical as in Eq. (6.79) but using a different error angle function as:

$$\chi_1 = (\pi - \theta_{sm1}) - \hat{\theta}_{sm1} \tag{6.101}$$

where $(\pi - \theta_{sm1})$ is the angle between the UAV velocity vector and the tangent line to the standoff circle around the target. In practice, an error angle χ_1 can be obtained by using difference between the engagement vector $\hat{\mathbf{t}}_1$ from Eq. (6.68) and UAV's current tangent vector $\mathbf{t}_1$. Differentiating Eq. (6.68) with respect to time gives:

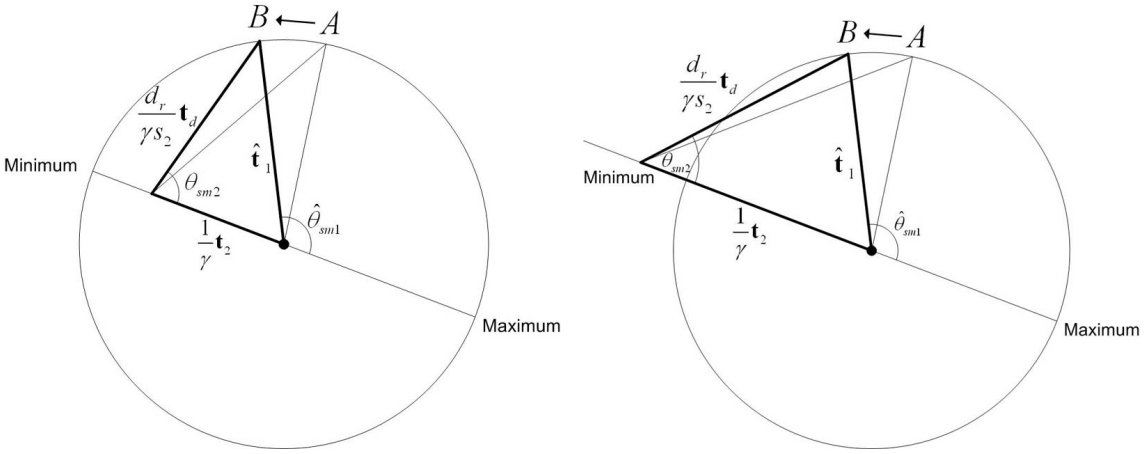
$$\dot{\hat{\mathbf{t}}}_1 = \frac{1}{\gamma}\left[\frac{d}{dt}\left(\frac{d_r}{s_2}\right)\mathbf{t}_d + \left(\frac{d_r}{s_2}\right)\dot{\mathbf{t}}_d\right] = \dot{\hat{\theta}}_1 \hat{\mathbf{n}}_1. \tag{6.102}$$

A geometric interpretation similar to the rendezvous problem is shown in Fig. 6.28. The same relation holds for the standoff tracking as for the rendezvous as:

$$\dot{\hat{\theta}}_{sm1} = \lambda_1(d_r/s_2, \theta_{sm2})\dot{\theta}_{sm2} \tag{6.103}$$

with:

$$-\left(1 + \frac{1}{\gamma}\right) \leq \lambda_1(d_r/s_2, \theta_{sm2}) \leq \left(1 + \frac{1}{\gamma}\right). \tag{6.104}$$



(a) Geometry $\gamma > 1$ (b) Geometry $\gamma < 1$

Figure 6.28: Geometric interpretation of standoff tracking solutions

The differential of r/s_2 can be obtained by differentiating Eq. (6.76) as:

$$\frac{d}{dt}\left(\frac{d_r}{s_2}\right) = \frac{-\sin(\theta_{sm2})\left(\frac{d_r}{s_2}\right)}{\left(\frac{d_r}{s_2}\right)-\cos(\theta_{sm2})}\dot{\theta}_{sm2} \tag{6.105}$$

In a similar way to the rendezvous case, by using θ_d, $\frac{d_r}{s_2}$, and θ_{sm2} instead of θ_s, $\frac{r}{s_2}$, and θ_{s2}, the guidance command to UAV, turn rate $\dot{\theta}_1$, can be modified to:

$$\dot{\theta}_1 = -\left(1+\frac{1}{\gamma}\right)\left|\dot{\theta}_d\right|\text{sign}(\chi_1) - k_\chi \chi_1 \tag{6.106}$$

where $k_\chi > 0$. The resulting time-derivative of Lyapunov candidate is:

$$\dot{L}_1 = \chi_1[-\left(1+\frac{1}{\gamma}\right)\left|\dot{\theta}_d\right|\text{sign}(\chi_1) + (1-\lambda_1)\,\dot{\theta}_d] - k_\chi \chi_1^2 \leq 0. \tag{6.107}$$

To avoid the chattering problem which results from the discontinuity of sign function, continuous saturation function could be applied. Note that the LOS rate $\dot{\theta}_s$ to the target is replaced by the tangent line rate $\dot{\theta}_d$ to the standoff circle in Eq. (6.106). As the UAV sensor generally measures a LOS angle, a range and their rates, a calculation of the tangent line rate $\dot{\theta}_d$ is required starting from the following relation as:

$$\theta_d = \theta_s \pm \theta_{d1}. \tag{6.108}$$

From Fig. 6.26, the following relation holds as:

$$\tan(\theta_{d1}) = \frac{d_m}{d_r}. \tag{6.109}$$

Differentiating both sides of the above equation gives

$$\frac{1}{\cos(\theta_{d1})^2}\dot{\theta}_{d1} = -\frac{d_m}{d_r^2}\dot{d}_r. \tag{6.110}$$

Rephrasing this yields:

$$\dot{\theta}_{d1} = -\frac{\cos(\theta_{d1})^2 d_m}{d_r^2}\dot{d}_r. \tag{6.111}$$

Substituting the relation $\cos(\theta_{d1}) = \frac{d_r}{r}$ gives:

$$\dot{\theta}_{d1} = -\frac{d_m}{r^2}\dot{d}_r. \tag{6.112}$$

Finally, substituting the above equation into the time-derivative of Eq. (6.108) yields:

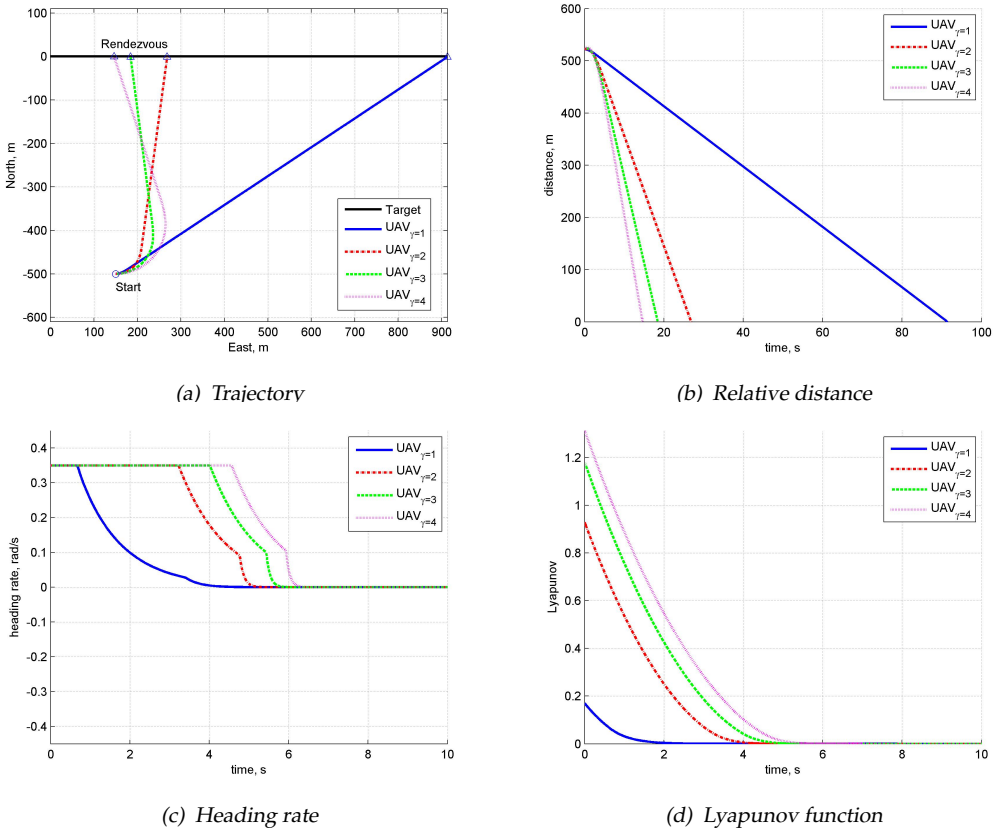
$$\dot{\theta}_d = \dot{\theta}_s \pm \frac{d_m}{r^2}\dot{d}_r. \tag{6.113}$$

Note that the solution involves the tangent line rate $\dot{d}_r$ to the standoff circle, not the LOS rate directly to the target position. This is practical since the range rate from the UAV to the target becomes zero and range measurement might be unobservable at the point of closest approach to the standoff orbit as the LOS gets normal to the tangent line.

In the case that the UAV is inside the standoff orbit, since a tangent line does not exist, the proposed guidance algorithm cannot be applied directly. Although several approaches can be suggested for this case such as keeping current speed until the UAV reaches standoff orbit or hybrid algorithm combining with other stable vector fields [143], this study uses modified control command $\dot{\theta}_m$ which exploits the condition of the UAV reaching the standoff orbit virtually whenever the UAV is inside the standoff orbit (i.e $d_m > r$) as:

$$\dot{\theta}_m = \dot{\theta}_{1t} \pm \Delta\dot{\theta}\tanh\left(\eta\frac{d}{d_m}\right) \tag{6.114}$$

where $d = r - d_m$, and $\eta > 0$ and $\Delta\dot{\theta}$ are control variables which adjust the convergence to the standoff orbit. $\dot{\theta}_{1t}$ represents temporary control command which makes the UAV track the target having the orbit radius of current distance from the target. As the UAV approaches to the desired standoff orbit, this modified control command gets closer to the original standoff tracking control command since d goes to zero.



(a) Trajectory (b) Relative distance

(c) Heading rate (d) Lyapunov function

Figure 6.29: Rendezvous of UAV to target with different speed ratios: $\gamma = 1, 2, 3,$ and 4

6.3.3 Numerical Simulations

6.3.3.1 Rendezvous and standoff tracking against constant-velocity target

This section carries out numerical examples using the proposed differential geometric guidance for rendezvous and standoff tracking of a UAV against a moving ground target. Here, it is assumed that the target velocity is ideally available, there is no actuator delay in the UAV controller, and w_{max} is set to be 0.35 rad/s. Firstly, Fig. 6.29 shows the simulation result of the rendezvous of the UAV to the target with different speed ratios between them. This simulation considers a UAV flying at different constant speeds of {10, 20, 30, 40} m/sec and a target traveling at a constant speed of 10 m/sec. Hence, γ =1, 2, 3, or 4 for this case. As can be seen in Fig. 6.29, the faster the UAV is compared to the target, the closer a rendezvous position gets to the initial position of the target. The UAV with a higher γ moves with a faster convergence time but consumes more control energy. For the case of

the UAV travelling at 10 m/sec ($\gamma = 1$) distant from the target, a parallel solution results in no convergence to rendezvous since $r/s_2 \rightarrow 0$ as shown in Fig. 6.30.

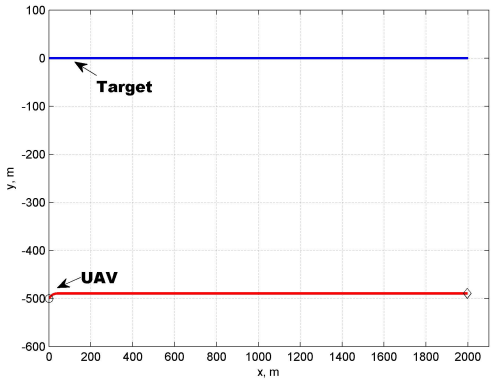
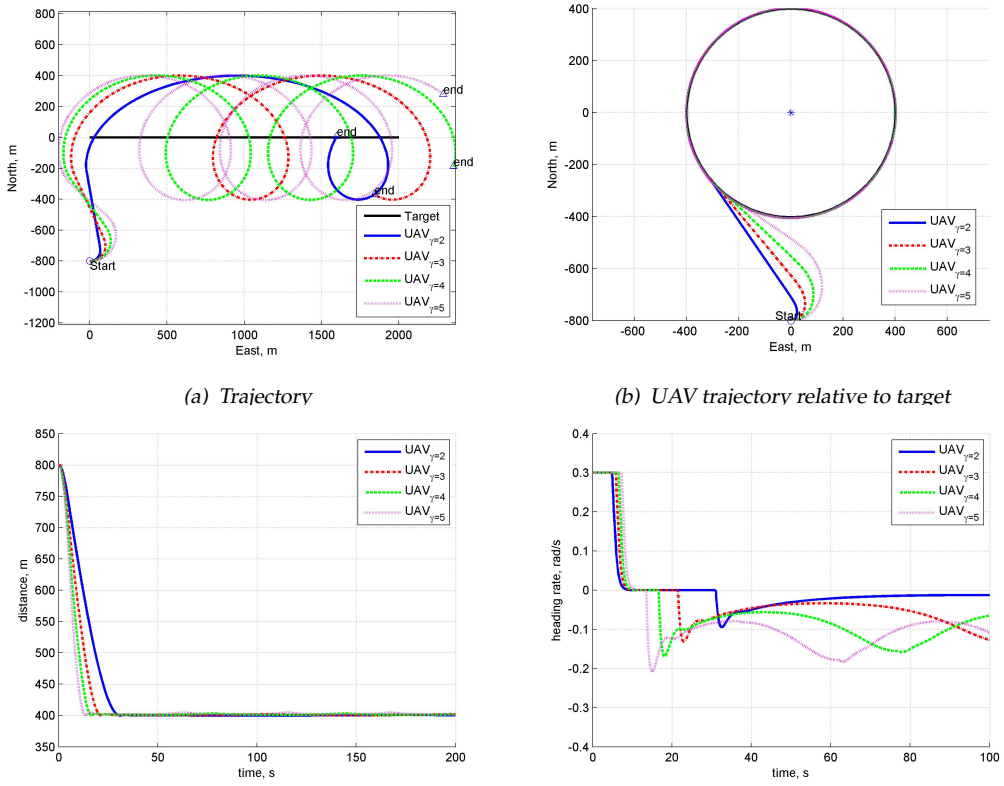


Figure 6.30: Parallel solution of rendezvous of the UAV to the target

The guidance law for standoff tracking is also applied to the scenario with the UAV flying different constant speeds of {20, 30, 40, 50} m/sec and a target travelling at a constant speed of 10 m/sec. Hence, γ =2, 3, 4, or 5 for this case. The cases of $\gamma = 1$ or lower values of γ are dropped since the UAV whose speed is not faster than the target is difficult to catch up with a standoff circle continuously. The required standoff distance from the UAV to the target d_m is set to be 300 m, w_{max} is set to be 0.3 rad/s, and the rotating direction of the UAV with respect to the target position is clockwise. The resultant trajectories for different γ are shown in Fig. 6.31. Figure 6.32 shows the standoff tracking for UAVs inside the standoff orbit with $d_m = 400$ m, $\Delta\dot{\theta} = 0.1$ and $\eta = 5$. As can be seen in the trajectories and relative distances of Fig. 6.31-6.32, the faster the UAV is compared to the target, the faster it converges to the standoff circle around the target. However, the UAV with a higher γ moves with consuming more control energy and aggressive manoeuvres.

6.3.3.2 Standoff tracking against realistic movement of ground vehicle

To verify the feasibility and benefits of the proposed approach in this section, a differential geometric guidance algorithm is applied to the same ground target tracking scenario as in Section 6.2.4. Target estimation is done by the EKF and



(a) Trajectory

(b) UAV trajectory relative to target

(c) Relative distance

(d) Heading rate

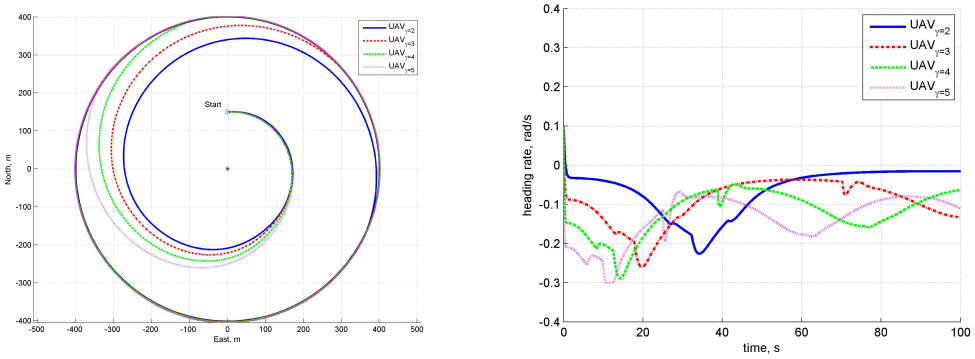
Figure 6.31: Standoff tracking of the UAV to the target with different speed ratios between UAV and target: $\gamma = 2, 3, 4,$ and 5

state-vector fusion as presented in Chapter 4. The phase-angle keeping is accomplished separately from the standoff orbit tracking by controlling the speed of UAVs as [14, 29]:

$$u_v = \pm k_v(\Delta\theta - \theta_d)d_m + v_d \tag{6.115}$$

where $k_v > 0$ is a control gain, $\Delta\theta$ is a difference of azimuth angles between UAVs relative to the target position, $\theta_d = \pi$ is a desired phase difference between UAVs, and v_d is a desired velocity command. In other words, by changing the speed of two vehicles accordingly moving on the same standoff orbit, desired angular separation can be achieved while avoiding collision each other as well as maximise the sensor coverage to the target. The setting of parameters needed for the proposed differential geometry guidance and estimation can be found in Table 7.1.

Figure 6.34(a) displays the absolute trajectories of UAVs and ground vehicle, and Fig. 6.34(b) displays the relative trajectories of UAVs with respect to the



(a) UAV trajectory relative to target

(b) Heading rate

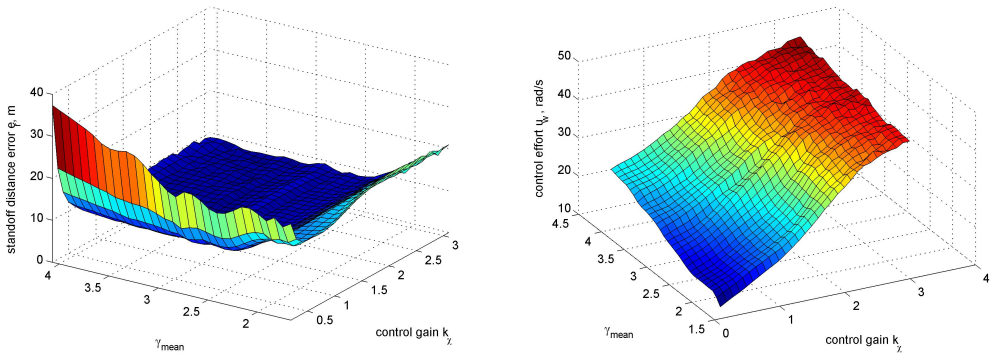
Figure 6.32: Standoff target tracking of the UAV inside standoff orbit, $\Delta\dot{\theta} = 0.1$ and $\eta = 5$

ground vehicle. As can be seen in Figs. 6.34(c)~(d), the proposed guidance shows a reasonably good tracking performance for both the standoff distance error and phase angle keeping between UAVs. Figures 6.34(e)~(f) show the control input histories of speed and turn rate. These are commanded control inputs, and as can be seen in the UAV dynamic model as Eq. (6.12), first-order time constants for considering actuator delay are used to respect dynamic constraints in the simulation providing smooth and realistic flying path.

Standoff tracking performance with different control gain k_χ and velocity ratio γ is investigated using the same scenario as above, but with a single UAV having a constant velocity during each simulation as shown in Fig. 6.33. In this figure, γ_{mean} represents the mean value of velocity ratio between the UAV and the moving target of each simulation, and it is changed by using different velocities of the UAV with the same ground target. Note that the variable γ is not tuned but in real-time computed in the guidance loop using the estimated target speed and the UAV's own one. As velocity ratio γ_{mean} and the control gain k_χ increase, the standoff distance error decreases except for the case of a small control gain. Since the target keeps changing its velocity, it is difficult for the UAV using a small control gain to track the target precisely, and it is becoming more difficult with increasing velocity of the UAV. On the other hand, the control effort obtained by integrating the time histories of $|u_w|$ tends to increase as the control gain and velocity ratio increase continuously. In short, these simulation results show the trade-off between tracking error and control effort, and facilitate system operators to define a requirement of the speed dominance of UAVs over the target depending on the mission specification.

Table 6.4: Simulation parameters

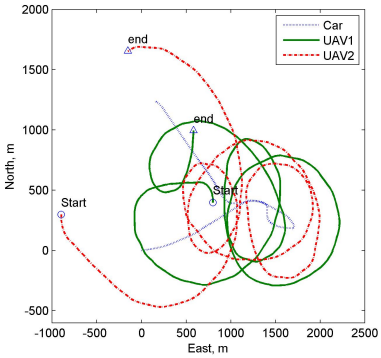
Parameter	Value	Unit
θ_d	π	rad
v_d	40	m/s
d_m	500	m
v_{max}	10	m/s
ω_{max}	0.2	rad/s
τ_v, τ_ω	1/3	sec
(k_v, k_χ)	(0.1, 1)	N/A
$(\sigma_{r1}, \sigma_{r2})$	(5, 7)	m
$(\sigma_{\phi 1}, \sigma_{\phi 2})$	(3.5, 2)	deg
$(\Delta\dot{\theta}, \eta)$	(0.1, 5)	N/A



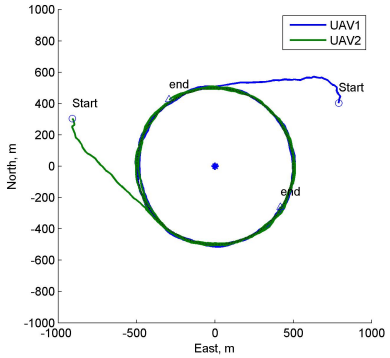
(a) Standoff distance error, $e_r = r - d_m$

(b) Control effort, u_w

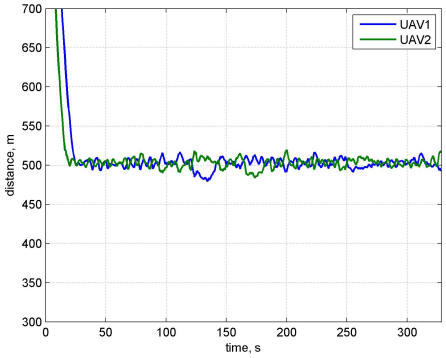
Figure 6.33: Standoff tracking performance with different control gain k_χ and velocity ratio $\bar{\gamma}$



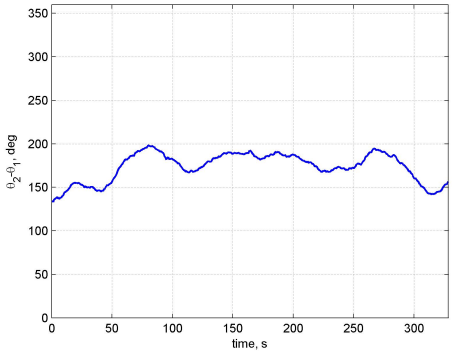
(a) Absolute trajectories of UAVs and target



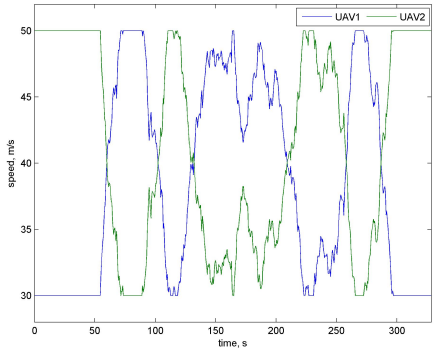
(b) Relative trajectories of UAVs with respect to target



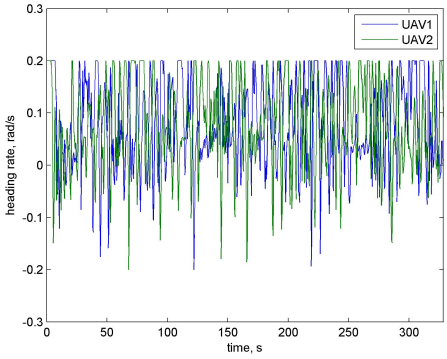
(c) Standoff distance error



(d) Phase angle difference of UAVs



(e) Control input of UAVs: u_v



(f) Control input of UAVs: u_ω

Figure 6.34: Standoff tracking simulation results

6.4 Nonlinear Model Predictive Coordinated Standoff Tracking

This section aims at developing a nonlinear model predictive coordinated standoff tracking (NMPCST) framework for both standoff distance and phase angle keeping by a pair UAVs. Typically, existing approaches for the standoff tracking problem such as vector field guidance uses the heading control for standoff-distance keeping and the speed control for phase keeping separately. Unlike this decoupled control structure, the NMPCST seeks to find the heading and velocity control inputs simultaneously using a coupled structure. The most valuable benefit of this approach is that it can get an optimal performance in terms of tracking and control efforts. In this way, the proposed optimal control formulation adds value by providing a basis for gain selection or parameter tuning for standoff tracking guidance algorithms.

The overall controller structure of the UAV formation considered in this study is fully decentralized in that each UAV optimises its controller based on the future state predictions of the pair UAV to cooperate with and the target to be tracked. This decentralized optimisation is performed after receiving only the control at the previous sampling, the current state, and the current target state/covariance estimation of the pair UAV via communication. For target trajectory prediction, this study uses an acceleration model for realistic target dynamics, which can consider the reasonable system noise covariance matrix reflecting the target's motion characteristics.

For antipodal standoff tracking, the phase keeping term is included in a performance index of optimisation framework as well as standoff-distance keeping. To simplify optimisation formulation and decrease computation burden, a new manipulation using inner product of position vectors of the UAVs with respect to the target position is proposed instead of using the relative phase-angle difference, which was used in most of the existing works [14, 29, 136]. To accomplish the safety under realistic situations, inequality constraints are also considered for collision avoidance between UAV members and control input saturations in the model predictive control scheme.

Numerical simulations with a pair of UAVs are done using realistic car tra-

jectory data in urban traffic to verify the feasibility and benefits of the proposed approach. Furthermore, in this model predictive approach that exploits future target trajectories, the prediction of the target movement plays an important role on tracking performance. Thus, this section investigates the effect of the improved estimation accuracy on the guidance performance by analysing both Lyapunov vector field and the proposed guidance.

6.4.1 Definition of Performance Index and Constraints

The geometry between the UAV, the pair UAV, and the ground target considered in this study is shown in Fig. 6.35. The basic aim of coordinated standoff target tracking is to maintain a distance $|\mathbf{r}|$ between the UAV and the target as well as a relative phase angle $\Delta\theta$ between UAVs to a desired value concurrently. Then,

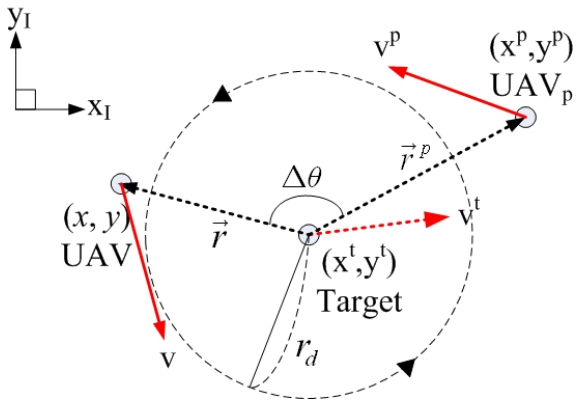


Figure 6.35: The geometry between the UAV, the pair UAV, and the ground target (The subscript k is deleted for convenience.)

coordinated standoff target tracking can be formulated as a nonlinear model predictive control problem to find a control input sequence $U_k = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$ that minimizes the following performance index.

$$J = \phi(\tilde{r}_N, \tilde{d}_N) + \sum_{k=0}^{N-1} L(\tilde{r}_k, \tilde{d}_k, \mathbf{u_k}) \tag{6.116}$$

$$\phi(\tilde{r}_N, \tilde{d}_N) = \frac{1}{2}(p_r \tilde{r}_N^2 + p_d \tilde{d}_N^2) \tag{6.117}$$

$$L(\tilde{r}_k, \tilde{d}_k, \mathbf{u_k}) = \frac{1}{2}\{q_r \tilde{r}_k + q_d \tilde{d}_k^2 + r_v(\frac{u_{vk} - v_0}{v_{max}})^2 + r_\omega(\frac{u_{\omega k} - \frac{v_0}{r_d}}{\omega_{max}})^2\} \tag{6.118}$$

where

$$\tilde{r}_k = \frac{r_d^2 - |\mathbf{r}_k|^2}{r_d^2} \tag{6.119}$$

$$\tilde{d}_k = \frac{\mathbf{r}_k^T \mathbf{r}_k^p + |\mathbf{r}_k||\mathbf{r}_k^p|}{r_d^2} \tag{6.120}$$

with $\mathbf{r}_k = C\mathbf{x}_k - C_t\mathbf{x}_k^t$ and $\mathbf{r}_k^p = C\mathbf{x}_k^p - C_t\mathbf{x}_k^t$ representing the relative vectors from the target position to the positions of the current UAV and its pair UAV, respectively. $C\mathbf{x}_k$, $C\mathbf{x}_k^p$ and $C\mathbf{x}_k^t$ represent the positions of the current UAV, the pair UAV, and the ground target which will be propagated according to time update equation of the Extended Kalman tracker, respectively. $C \in R^{2\times5}$ and $C_t \in R^{2\times6}$ is simply represented by the definition of state variables: $\mathbf{x}_k = (x_k, y_k, \psi_k, v_k, \omega_k)^T$, $\mathbf{x}_k^p = (x_k^p, y_k^p, \psi_k^p, v_k^p, \omega_k^p)^T$, and $\mathbf{x}_k^t = (x_k^t, \dot{x}_k^t, \ddot{x}_k^t, y_k^t, \dot{y}_k^t, \ddot{y}_k^t)^T$:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.121}$$

$$C_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{6.122}$$

Also, r_d is a desired standoff distance from the UAVs to the ground target position, N is the length of receding horizon, v_0 is a nominal speed of UAVs, and $\frac{v_0}{r_d}$ is a nominal angular velocity. p_r, p_d, q_r, q_d, r_v, and r_ω are constant weighting scalars.

In Eq. (6.120), $\tilde{d}_k$ is derived from the inner product of $\mathbf{r}_k$ and $\mathbf{r}_k^p$ as:

$$< \mathbf{r}_k, \mathbf{r}_k^p >= \mathbf{r}_k^T \mathbf{r}_k^p = |\mathbf{r}_k||\mathbf{r}_k^p| \cos \triangle\theta_k \tag{6.123}$$

where $\Delta\theta_k = |\theta_k^p - \theta_k|$ with the phase angles of UAV positions with respect to the current target location, $\theta_k = \cos^{-1} \frac{<\mathbf{r}_k, \hat{i}>}{|\mathbf{r}_k|}$ and $\theta_k^p = \cos^{-1} \frac{<\mathbf{r}_k^p, \hat{i}>}{|\mathbf{r}_k^p|}$ ($\hat{i}$ is a x-direction unit vector.). If the phase difference $\Delta\theta_k$ is ideally maintained as π radian for antipodal tracking of a pair of UAVs, the above equation is rearranged since $\cos\pi = -1$ as:

$$\mathbf{r}_k^T \mathbf{r}_k^p + |\mathbf{r}_k||\mathbf{r}_k^p| = 0. \tag{6.124}$$

Therefore, the left-hand side of the above equation can be set as $\tilde{d}_k$ in the performance index to be minimized for maintaining the phase angle between a pair of UAVs. The merit of using this inner product concept rather than $\tilde{\theta}_k = \triangle\theta_k - \pi$ is

two fold: the first is that practically it can minimize angular manipulations which might bring about unnecessary switching and laborious logic development at the boundary of angle range $(-\pi, \pi)$. The other is the unit of the phase-keeping term becomes the same as the first term in Eqs. (6.117)~(6.118) contributing to maintaining a standoff distance from the target. This is important for accomplishing the fast convergence of the optimisation algorithm and also for avoiding unnecessary tuning of weighting parameters.

The dynamics of the UAVs in Eq. (6.15) is expressed as an equality constraint as:

$$f_d(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1} = 0. \tag{6.125}$$

The collision avoidance requirement between the UAV members is adopted as an inequality constraints as well as admissible control input ranges.

$$S_v(\mathbf{u}_k) = \frac{|u_{1k} - v_0| - v_{max}}{v_{max}} \leq 0 \tag{6.126}$$

$$S_\omega(\mathbf{u}_k) = \frac{|u_{2k}| - \omega_{max}}{\omega_{max}} \leq 0 \tag{6.127}$$

$$S_c(\mathbf{x}_k) = \frac{r_c - |C(\mathbf{x}_k - \mathbf{x}_k^p)|}{r_c} \leq 0 \tag{6.128}$$

where r_c is a safe distance between the UAVs to prevent collision.

6.4.2 NMPCST Algorithm

The augmented performance index is defined with incorporating the equality and inequality constraints discussed in the previous section as:

$$\begin{aligned} J_a = \phi(\tilde{r}_N, \tilde{d}_N) + \sum_{k=0}^{N-1} [L(\tilde{r}_k, \tilde{d}_k, \mathbf{u}_k) + \lambda_{k+1}^T \{f_d(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1}\} \\ + \frac{1}{2}\mu_v l_{vk} S_v^2(\mathbf{u}_k) + \frac{1}{2}\mu_\omega l_{\omega k} S_\omega^2(\mathbf{u}_k) + \frac{1}{2}\mu_c l_{ck} S_c^2(\mathbf{x}_k)] \end{aligned} \tag{6.129}$$

where λ_k is a Lagrange multiplier, and μ_v, μ_ω, and μ_c are penalty function parameters. l_{vk}, $l_{\omega k}$, and l_{ck} are defined to avoid unnecessary computation for satisfying inequality constraints as [153]:

$$l_{*k} = \begin{cases} 0, & S_* \leq 0 \\ 1, & S_* > 0 \end{cases}. \tag{6.130}$$

To facilitate the derivation of optimal control law, let us define a Hamiltonian as:

$$M_k \triangleq L(\tilde{r}_k, \tilde{d}_k, \mathbf{u}_k) + \lambda_{k+1}^T f_d(\mathbf{x}_k, \mathbf{u}_k) + \frac{1}{2}\mu_v l_{vk} S_v^2(\mathbf{u}_k) + \frac{1}{2}\mu_\omega l_{\omega k} S_\omega^2(\mathbf{u}_k) + \frac{1}{2}\mu_c l_{ck} S_c^2(\mathbf{x}_k). \quad (6.131)$$

The variation of the augmented performance index is represented as:

$$dJ_a = (\frac{\partial \phi(\tilde{r}_N, \tilde{d}_N)}{\partial \mathbf{x}_N} - \lambda_N^T) d\mathbf{x}_N + \sum_{k=1}^{N-1} [(\frac{\partial M_k}{\partial \mathbf{x}_k} - \lambda_k^T) d\mathbf{x}_k + \frac{\partial M_k}{\partial \mathbf{u}_k} d\mathbf{u}_k] + \frac{\partial M_0}{\partial \mathbf{x}_0} d\mathbf{x}_0 + \frac{\partial M_0}{\partial \mathbf{u}_0} d\mathbf{u}_0. \quad (6.132)$$

By selecting the Lagrange multiplier as:

$$\lambda_N^T = \frac{\partial \phi(\tilde{r}_N, \tilde{d}_N)}{\partial \mathbf{x}_N} \quad (6.133)$$

$$\lambda_k^T = \frac{\partial M_k}{\partial \mathbf{x}_k} \quad \text{for } k = \mathrm{N} - 1, \ldots, 0, \quad (6.134)$$

the variation of J_a is changed to:

$$dJ_a = \sum_{k=0}^{N-1} \frac{\partial M_k}{\partial \mathbf{u}_k} d\mathbf{u}_k + \lambda_0^T d\mathbf{x}_0. \quad (6.135)$$

The right-hand side of Eq. (6.133) is derived from Eq. (6.117) as:

$$\frac{\partial \phi(\tilde{r}_N, \tilde{d}_N)}{\partial \mathbf{x}_N} = p_r \tilde{r}_N \frac{\partial \tilde{r}_N}{\partial \mathbf{x}_N} + p_d \tilde{d}_N \frac{\partial \tilde{d}_N}{\partial \mathbf{x}_N}. \quad (6.136)$$

The right-hand side of Eq. (6.134) is derived from Eq. (6.131) as:

$$\frac{\partial M_k}{\partial \mathbf{x}_k} = q_r \tilde{r}_k \frac{\partial \tilde{r}_k}{\partial \mathbf{x}_k} + q_d \tilde{d}_k \frac{\partial \tilde{d}_k}{\partial \mathbf{x}_k} + \lambda_{k+1}^T \frac{\partial f_d(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} + \mu_c l_{ck} S_c(\mathbf{x}_k) \frac{\partial S_c(\mathbf{x}_k)}{\partial \mathbf{x}_k}. \quad (6.137)$$

Jacobians in Eqs. (6.136)~(6.137) are derived from the definitions discussed in the previous section.

$$\frac{\partial \tilde{r}_k}{\partial \mathbf{x}_k} = \frac{1}{r_d^2} \begin{bmatrix} -2(x_k - x_k^t) & -2(y_k - y_k^t) & 0 & 0 & 0 \end{bmatrix} \quad (6.138)$$

$$\frac{\partial \tilde{d}_k}{\partial \mathbf{x}_k} = \frac{1}{r_d^2} \begin{bmatrix} (x_k^p - x_k^t) + \frac{|\mathbf{r}_k^p|}{|\mathbf{r}_k|}(x_k - x_k^t) & (y_k^p - y_k^t) + \frac{|\mathbf{r}_k^p|}{|\mathbf{r}_k|}(y_k - y_k^t) & 0 & 0 & 0 \end{bmatrix} \quad (6.139)$$

$$\frac{\partial f_d(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} 1 & 0 & -T_s v_k \sin\psi_k & T_s \cos\psi_k & 0 \\ 0 & 1 & T_s v_k \cos\psi_k & T_s \sin\psi_k & 0 \\ 0 & 0 & 1 & 0 & T_s \\ 0 & 0 & 0 & 1 - \frac{1}{\tau_v} T_s & 0 \\ 0 & 0 & 0 & 0 & 1 - \frac{1}{\tau_\omega} T_s \end{bmatrix} \quad (6.140)$$

$$\frac{\partial S_c(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \frac{1}{r_c^2} \begin{bmatrix} -\frac{x_k - x_k^p}{|\mathbf{r}_k - \mathbf{r}_k^p|} & -\frac{y_k - y_k^p}{|\mathbf{r}_k - \mathbf{r}_k^p|} & 0 & 0 & 0 \end{bmatrix} \quad (6.141)$$

Now substituting $d\mathbf{u}_k$ into Eq. (6.135) with the following equation which is basically a gradient method to minimize M_k

$$d\mathbf{u}_k = -\Delta_k \frac{\partial M_k}{\partial \mathbf{u}_k}^T \tag{6.142}$$

gives the following decreasing variation of J_a.

$$dJ_a = -\sum_{k=0}^{N-1} \Delta_k \frac{\partial M_k}{\partial \mathbf{u}_k} \frac{\partial M_k}{\partial \mathbf{u}_k}^T + \lambda_0^T d\mathbf{x}_0. \tag{6.143}$$

Therefore, the control input can be updated using Eq. (6.142) as:

$$\mathbf{u}_k^{i+1} = \mathbf{u}_k^i - \Delta_k \frac{\partial M_k}{\partial \mathbf{u}_k}^T \quad \text{for } k = 0, \ldots, \mathrm{N}-1 \tag{6.144}$$

where i is the index of iteration, and Δ_k is a step size. $\frac{\partial M_k}{\partial \mathbf{u}_k}$ is defined as:

$$(\frac{\partial M_k}{\partial \mathbf{u}_k})^T = \begin{bmatrix} \frac{r_v}{v_{max}^2}(u_{vk} - v_0) \\ \frac{r_\omega}{\omega_{max}^2}(u_{\omega k} - \frac{v_0}{r_d}) \end{bmatrix} + \lambda_{k+1}^T \frac{\partial f_d(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{u}_k} + \begin{bmatrix} \frac{\mu_v}{v_{max}^2} l_{vk} S_v(\mathbf{u}_k) \mathrm{sign}(u_{vk} - v_0) \\ \frac{\mu_\omega}{\omega_{max}^2} l_{\omega k} S_\omega(\mathbf{u}_k) \mathrm{sgn}(u_{\omega k} - \frac{v_0}{r_d}) \end{bmatrix} \tag{6.145}$$

where

$$\frac{\partial f_d(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{u}_k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \frac{1}{\tau_v} T_s & 0 \\ 0 & \frac{1}{\tau_\omega} T_s \end{bmatrix}. \tag{6.146}$$

6.4.3 Online Optimisation via Decentralized Structure

The online optimisation is run under the decentralized structure at each sampling T_s as shown in Fig. 6.36. When the GMTI measurement on the target composed of the relative range and azimuth with respect to each UAV position comes in, each UAV performs the localization of the target by the Extended Kalman tracker. Then, control (U_{k-1})/state ($\mathbf{x}_k$) of the UAVs and those state ($\mathbf{x}_{k|k}^t$)/covariance ($P_{k|k}$) estimation information of the target are shared via communication between the UAVs. This study assumes all the required information can be shared between the UAVs via communications within a single sampling. If communications delay

happens or optimisations are performed at slightly different instances between in a real world, the performance of the proposed approach might be degraded. However, our decentralized structure is advantageous in the sense that each UAV performs its own optimisation individually using as best information as possible with a reduced computation burden compared to a centralized approach. Also, the proposed decentralized strategy can improve the probability of mission completion because the remaining pair UAV can continue the standoff tracking mission in spite of a failure of one UAV. Then, each UAV carries out the state-vector fusion using its own estimation and the information from its pair UAV as shown in Eq. (4.62) in order to enhance the tracking accuracy against the moving target, which is then used as the initial value for propagating the target's state using the time update equation of the Extended Kalman tracker in Eq. (4.28). Following this, the state of the pair UAV is propagated using Eq. (6.15) with the received pair-UAV information: the current state information and the optimal control over the receding horizon computed at the previous sampling. Now all the external information is ready for the online optimisation of nonlinear model predictive control.

The detailed online optimisation has the following procedures: the initialization of control input over a receding horizon, state propagation (equality constraint), the computation of an augmented performance index, the finding of a Lagrange multiplier by backward integration, the computation of a Jacobian matrix of Hamiltonian M_k with respect to the control input, and the update of control inputs. As a practical and efficient way, the initial guess of control inputs over the receding horizon at the first sampling $k = 0$ can use a desired nominal speed of UAV and a desired angular velocity for converging to the standoff circle as:

$$U_0^0 = \{\mathbf{u}_0, \mathbf{u}_0, \ldots, \mathbf{u}_0\} \in \mathbb{R}^{2\times N} \tag{6.147}$$

where $\mathbf{u}_0 = (u_{v0}, u_{\omega 0})^T$, $u_{v0} = v_d$, and $u_{\omega 0} = \frac{v_d}{r}$. Then, if $k > 0$, the initialization can be done by shifting the optimised control history from the previous sampling $U_{k-1} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-2}, \mathbf{u}_{N-1}\}$ as:

$$U_k^0 = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{N-1}, \mathbf{u}_{N-1}\}. \tag{6.148}$$

The detailed optimisation routine of the proposed approach is shown in Fig. 6.36.

Figure 6.36: Decentralized structure of nonlinear model predictive coordinated standoff tracking

current UAV	Communications	Pair UAV
Start k-th sampling time		
GMTI measurement acquisition		GMTI measurement acquisition
EKF Target localization [Eqs.(4.18)-(4.40)]	$\xrightarrow{\hat{\mathbf{x}}^{tp}_{k\|k}, P^{p}_{k\|k}}$	Target localization by EKF
	$\xleftarrow{\hat{\mathbf{x}}^{tp}_{k\|k}, P^{p}_{k\|k}}$	
State-vector fusion [Eq.(4.62)]		State-vector fusion
Target state propagation [Eq.(4.18)]	$\xrightarrow{\mathbf{x}_k, U_{k-1}}$	Target state propagation
	$\xleftarrow{\mathbf{x}^{p}_{k}, U^{p}_{k-1}}$	
Pair-UAV state propagation [Eq.(6.15)]		Pair-UAV state propagation
$i = 1$		$i = 1$
while $dJ_a > \epsilon$		while $dJ_a > \epsilon$
If $i == 1$,		If $i == 1$,
U^0_k Initialization [Eqs.(6.147)-(6.148)]		U^0_k Initialization
end		end
Self state propagation [Eq.(6.15)]		Self state propagation
J_a computation [Eq.(6.129)]		⋮
λ_k computation [Eqs.(6.133)-(6.134)]		⋮
Computation of $\frac{\partial M_k}{\partial \mathbf{u}_k}$ [Eq.(6.145)]		⋮
If $dJ_a < 0$: $U^i_k = U^{i-1}_k - \Delta_k \frac{\partial M_k}{\partial \mathbf{u}_k}$		⋮
else: reduce step size Δ_k		⋮
end		end
$i = i + 1$		$i = i + 1$
end		end
Start $k+1$-th sampling time		
⋮		⋮

6.4.4 Numerical Simulations

To verify the feasibility and benefits of the proposed approach in this study, the same tracking scenario used in the Section 6.3.3.2 were tested for both the LVFG and NMPCST. The setting of parameters needed for the NMPCST (Nonlinear Model Predictive Coordinated Standoff Tracking) and LVFG can be found in Table 7.1.

Table 6.5: Simulation parameters

Parameter	Value	Unit
T_s	0.5	sec
α	0.6	N/A
σ_a	0.66	m/s^2
θ_d	π	rad
v_0, v_d	40	m/s
r_d, r_c	500, 30	m
v_{max}	10	m/s
ω_{max}	0.2	rad/s
τ_v, τ_ω	1/3	sec
N	4 (equivalent to 2 secs)	N/A
$(p_r, p_d, q_r, q_d, r_v, r_\omega)$	$(2e5, 5e6, p_r/N, p_d/N, 1e2, 5e1)$	N/A
μ_v, μ_ω, μ_c	$1e3$	N/A
(k_v, k_ω)	(0.12, 2)	N/A
$(\sigma_{r1}, \sigma_{r2})$	(5, 7)	m
$(\sigma_{\phi 1}, \sigma_{\phi 2})$	(3.5, 2)	deg

Figure 6.37 displays the relative trajectories of the UAVs with respect to the ground vehicle. As can be seen in Figs. 6.37~6.38 and Table 6.6, the mean error of standoff-distance keeping by the NMPCST is much lower than that of LVFG (which was tuned at best by trial and error procedure) even though its control consumption on rate of heading change is slightly higher than that of LVFG. This results from the fact that the NMPCST optimises its control input by taking account of the propagated histories of the states of target and UAVs over the receding horizon. The control consumptions of u_v and u_ω in Table 6.6 are computed by integrating the time histories of $|u_v - v_0|$ and $|u_\omega|$, respectively. Although the mean

error of phase-angle keeping by the LVFG could be improved by increasing the gain k_v, its value is set as 0.12 so that the LVFG has a similar level of control input consumption for fair comparison. As can be seen in Table 6.6, the NMPCST has a slightly lower mean error for phase-angle keeping than the LVFG with a similar level of speed consumption. This is beneficial to fixed-wing UAVs since they generally consume higher fuel when controlling their forward speed. Intuitively, this merit results from the fact that part of heading rate control as well as speed change by the NMPCST contributes on keeping a phase difference to meet the inner product relation of position vectors as in Eq. (6.124) during optimisation. Figures 6.40~6.41 show the control input histories of speed and heading-rate change of both approaches. Note that high frequency control inputs are required to be followed by UAVs for both velocity and turning rate, which is hard to achieve in practice. Even though actuator time delay (τ_v, τ_ω) in a UAV kinematic model is used in order to simulate this effect, more detailed control requirements will be investigated in future work.

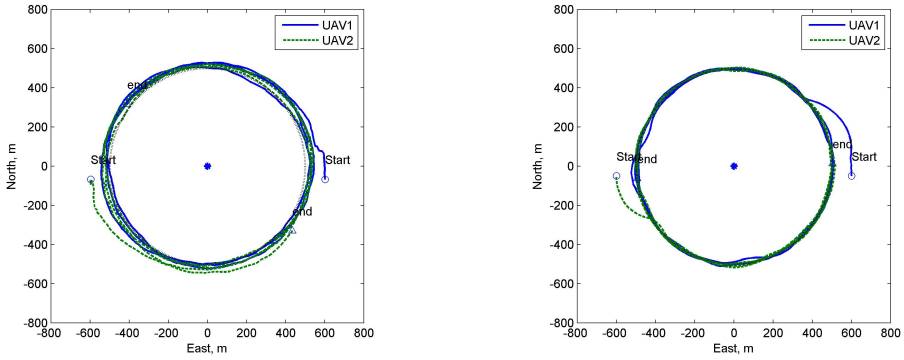
Table 6.6: Performance comparison between LVFG and NMPCST

Parameter	LVFG		NMPCST	
standoff distance error (m)	mean±std	min/max	mean±std	min/max
	22.90±8.65	2.61/48.56	5.38± 4.14	0.09/25.90
phase keeping error (deg)	mean±std	min/max	mean±std	min/max
	7.62±4.52	0.03/20.17	7.28± 4.21	0.01/17.97
u_v control consumption (m/s)	18.79e2		17.33e2	
u_ω control consumption (rad/s)	28.61		29.30	

Besides, to evaluate effect of the improved estimation accuracy on the guidance performance, another numerical simulation is performed by comparing tracking guidance performance using filtering techniques with the road information, as presented in Chapter 4. The measurement noise is set to significant values in this simulation, in order to show the benefit of constrained filtering more clearly as:

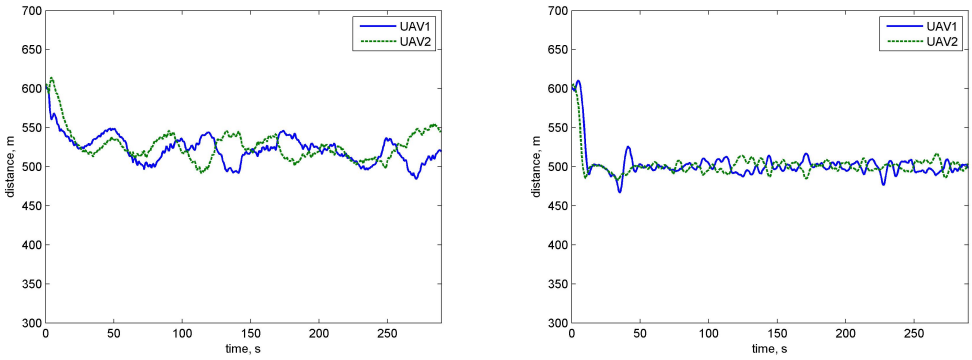
$$\text{UAV1:} \quad (\sigma_{r1}, \sigma_{\phi 1}) = (20\ m, 7\ deg) \tag{6.149}$$

$$\text{UAV2:} \quad (\sigma_{r2}, \sigma_{\phi 2}) = (30\ m, 5\ deg) \tag{6.150}$$



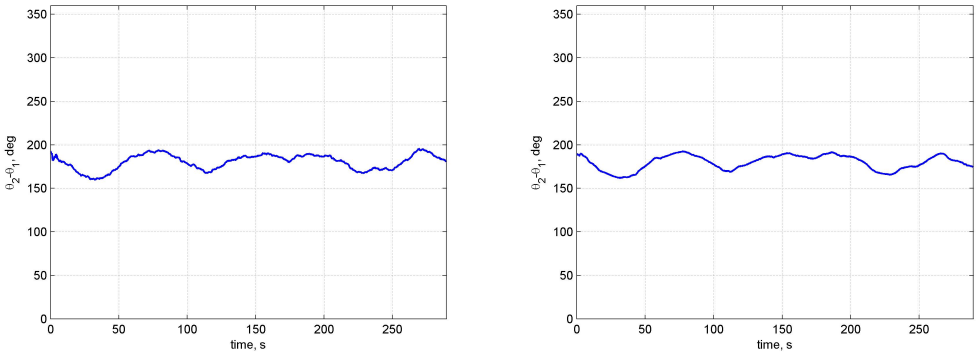
(a) LVFG (b) NMPCST

Figure 6.37: Relative trajectories of UAVs with respect to ground vehicle



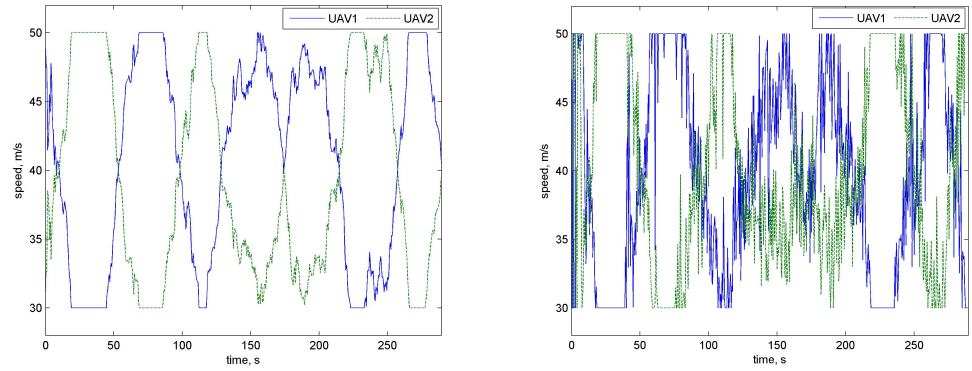
(a) LVFG (b) NMPCST

Figure 6.38: Standoff distance from UAVs to ground vehicle



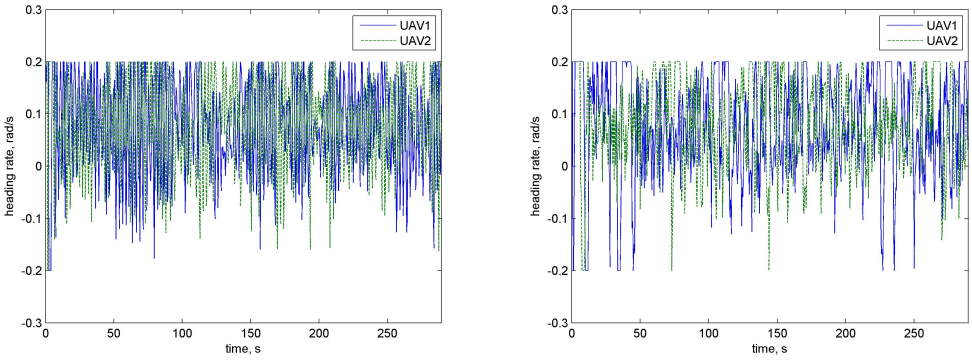
(a) LVFG (b) NMPCST

Figure 6.39: Phase angle difference of UAVs with respect to ground vehicle



(a) LVFG (b) NMPCST

Figure 6.40: Control input of UAVs: u_v



(a) LVFG (b) NMPCST

Figure 6.41: Control input of UAVs: u_ω

Table 6.7 shows mean tracking error in position and velocity among different filtering methods. Apparently, using the state-vector fusion of two UAVs shows better performance than the one using only single UAV, and the road-constrained ECUKF with state-vector fusion provides the best estimation accuracy. Figures 6.42~6.43 display estimated position and velocity of a ground target using the state-vector fusion based on the EKF and the road-constrained ECUKF, respectively. Table 6.8 compares tracking guidance performance for standoff distance and phase keeping between LVFG and NMPCST using either the EKF or the road-constrained ECUKF. It is worthwhile noting that the performance improvement of NMPCST with changing estimation method from the EKF to the ECUKF is more remarkable than that of LVFG, since NMPCST uses predicted target's information to a certain future time explicitly.

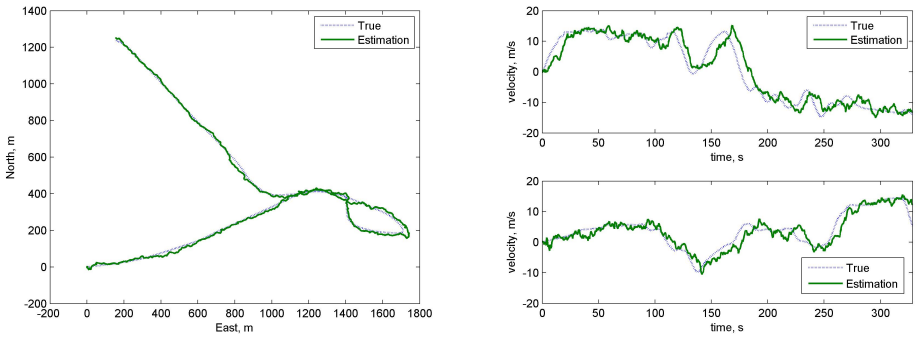
Table 6.7: Performance comparison with different estimation methods

Mean error	Single UAV		Multiple UAVs (State-vector fusion)			
	Unconstrained		Unconstrained		Road-constrained	
	$EKF_{UAV,1}$	$EKF_{UAV,2}$	EKF_{multi}	UKF_{multi}	$MAEKF_{multi,c}$	$ECUKF_{multi,c}$
Position (m)	18.2612	18.9317	14.4422	14.1452	8.5729	8.1752
Velocity (m/s)	3.3433	3.5795	3.1123	3.1166	2.1712	2.1565

Table 6.8: Tracking performance with different estimation methods

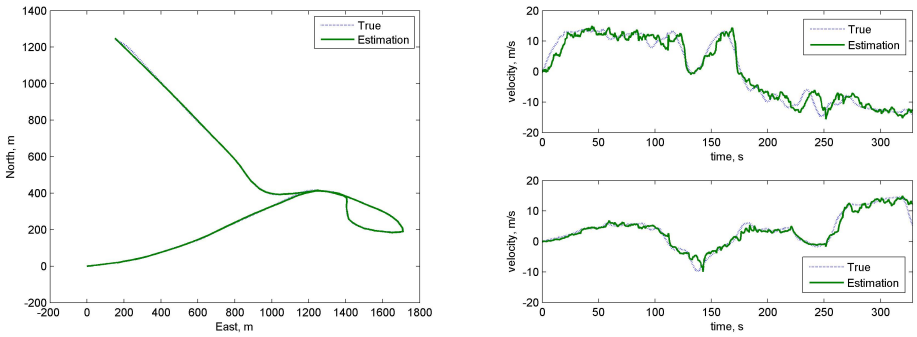
Mean error	LVFG		NMPCST	
	EKF_{multi}	$ECUKF_{multi,c}$	EKF_{multi}	$ECUKF_{multi,c}$
Standoff distance (m)	15.8830	13.0514	14.8521	9.2001
Phase keeping (deg)	13.1350	13.0020	13.3673	11.3492

The comparison performed in this simulation does not assert that the NMPCST can always get more optimal performance than the LVFG in terms of mission performance or fuel efficiency because the LVFG does not aim only precision in tracking. However, this simulation results implies that the proposed nonlinear model predictive approach can add a beneficial value to the coordinated standoff tracking problem by providing a basis for gain selection or parameter tuning. The mean computation time of each UAV's NMPCST routine at every sampling is approximately 0.1 second when using Intel CPU E5440 2.83GHz and 3GB RAM.



(a) Trajectory *(b) x and y velocity*

Figure 6.42: State-vector fusion results based on the EKF



(a) Trajectory *(b) x and y velocity*

Figure 6.43: State-vector fusion results based on the road-constrained ECUKF

6.5 Discussions

This chapter proposed four different guidance algorithms for coordinated standoff tracking of an unknown moving ground target. Unlike the existing algorithms, this study explicitly incorporates target estimation into standoff tracking guidance with a realistic car trajectory. Besides, the effect of improved estimation accuracy with sensor fusion techniques using multiple UAVs on guidance performance is extensively investigated. The following summarises the advantages and disadvantages/limitations of each method.

Two-phase orbit approach based on path shaping

- This path shaping approach with a constant speed would be desirable for a mission duration point of view.
- This requires less communication between UAVs to achieve simultaneous arrival to the standoff orbit, compared to the other guidance approaches in which information needs to be shared continuously.
- This would be difficult to be applied to an unknown moving target tracking.

Decentralised vector field guidance using adaptive sliding mode control

- Adaptive sliding mode control concept can significantly reduce the effect of unmodelled dynamics and disturbance in the UAV autopilot.

- Either velocity or orbit radius change can lead to successful standoff tracking. In case that frequent velocity change is undesirable or unattainable, the angular separation can be achieved without velocity control but with a bounded error.
- Decentralised approach can directly combine the estimation and guidance considering different information/communication structures.

Standoff tracking guidance based on differential geometry

- This approach brings advantages along with its inherent simplicity: rigorous stability, explicit use of a target velocity, and tuning parameter reduction.
- In case that UAVs are inside the standoff orbit, an ad-hoc method is currently being used.
- Coordination strategy based on differential geometry needs to be developed.

Nonlinear model predictive coordinated standoff tracking

- NMPCST guidance can provide an optimal performance in terms of tracking as well as control efforts.
- This method depends heavily on the UAV model and target estimation accuracy to propagate commands.
- High frequency control inputs are required to be followed by UAVs for both velocity and turning rate, which is hard to achieve in practice.
- This would require significant computation power to perform a real-time optimisation.

Note that this chapter has addressed the problem of a single target tracking only. However, in reality, several distinct or a group of suspicious targets could be observed in geographically close areas. This requires a more complicated consideration on a standoff tracking problem involving a task allocation between UAVs and targets as well as tracking of clustered targets. In this regard, coordinated standoff tracking of groups of moving targets using multiple UAVs will be considered in the next chapter.

Chapter 7

Coordinated Standoff Tracking of Groups of Moving Targets

In the previous chapter, a circular standoff tracking concept is introduced to closely orbit around the target maintaining sensor coverage and remaining outside a critical threat range. These circular flights are recommended for various target tracking applications as for each UAV the maximum altitude flight ensures the maximum visibility and the minimum radius turn keeps the minimum distance to the target at the maximum altitude [28]. Standoff target tracking using cooperative UAVs is also proposed by distributing a team of UAVs on a standoff orbit when the target vehicle is uncooperative, or is highly agile. This cooperative standoff tracking of a moving target using multiple UAVs can provide better estimation accuracy with sensor/data fusion through communication between UAVs [30, 154] as well as more robust tracking performance in case of temporary sensing failure or LOS block by obstacles for one of UAVs [28].

However, aforementioned previous research on standoff tracking has focused on the single target tracking problem. In case that multiple moving ground vehicles are identified as targets of interest from reconnaissance or surveillance systems within the ground road traffic, it is required to develop strategies on how to deploy multiple UAVs to persistently follow them. Although various different methodologies have been developed for the multi-target tracking using multiple ground [35, 36] or aerial vehicles [37, 38], there is relatively little research on multiple or group target tracking in the context of cooperative standoff tracking considering uncertain dynamic environments and UAV sensing capability.

Therefore, this chapter proposes a coordinated standoff tracking methodology of groups of moving targets using multiple UAVs. In order to track a group of targets using a sensor with the limited field-of-view (FOV), the vehicle should be positioned as close as possible to multiple targets to obtain better estimation accuracy and far enough to keep the group of targets within its FOV. For this, amongst many standoff guidance algorithms explained in the previous chapter, the Lyapunov vector field guidance approach is applied, which produces stable

convergence to a circling limit cycle behaviour [14, 151]. The objective of this study is to develop an active sensing/guidance algorithm to maximise information or accuracy of estimation of the targets as well as persistently keep all of them (or as many as possible) within the view of multiple UAVs considering physical (turning radius and speed) and sensing (FOV and range) constraints.

The main contributions of this chapter are fourfold. Firstly, this study proposes a new coordinated group target tracking method in the context of standoff tracking by defining a variable standoff orbit to be followed, which can keep all targets within the FOV of the UAV sensor considering uncertainty of estimated target information. Secondly, a new feedforward term is computed in the guidance command considering a variable standoff distance compared to a single target tracking case having a fixed standoff distance. Moreover, convergence of the vector field to the variable standoff orbit is analysed and enhanced by adjusting radial velocity using two active measures associated with vector field generation. Thirdly, for multiple group target surveillance by multiple UAVs, a two-phase approach is proposed as a sub-optimal solution for an NP-hard problem: (i) multiple targets are clustered using K-means clustering algorithm, and UAVs are assigned to the appropriate target group which maximises information defined by the Fisher information matrix (FIM), and then (ii) cooperative standoff group tracking is performed with online local replanning including target handoff and discard from the group according to sensing capability and vector field convergence. Lastly, localisation sensitivity to the group of targets is investigated for different angular separations between UAVs and sensing configurations as a basis of future optimal separation scheme.

7.1 Problem Formulation

It is first assumed that lateral and longitudinal dynamics of the UAV can be decoupled as in conventional fixed wing aircraft. Therefore, a two-dimensional space is considered for the UAV flying at a constant altitude. In this study, it is also assumed that initial target information is given by other sources such as search and monitoring UAV as presented in Chapter 5 and sensor can point at the group centre using a gimbal system. Note that data association for multiple targets and communication between UAVs are not considered in this study. UAV team

members share a known global coordinate system such as GPS (Global Positioning System) for their own and target's position. The concept of the standoff tracking problem of groups of moving targets using multiple UAVs is illustrated in Fig. 7.1. The standoff orbit for each group followed by UAVs needs to be changed in terms of size and location according to the dispersion of the moving targets so that all targets can be inside the FOV of UAVs.

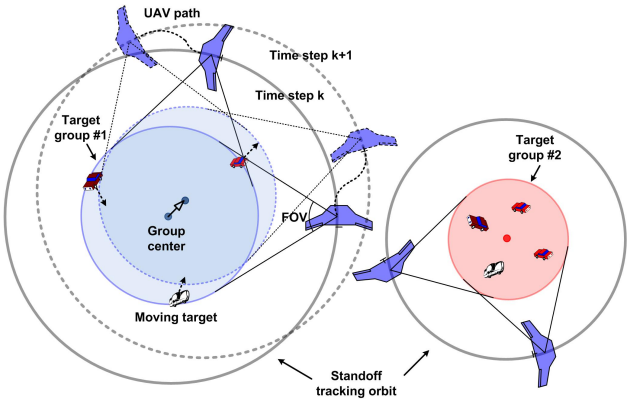


Figure 7.1: Illustration of the standoff tracking of groups of moving targets using multiple UAVs considering sensing constraints

For the target localisation, the same UAV, ground target, and sensor model as in Chapter 4 is used with the extended Information filter (EIF). Having estimated all available target's information, the information on the centre of a target group is also estimated using the same target model as for an individual target, position measurements of geometric centroid for targets in the group, and linear Kalman filter providing $\mathbf{x}_k^{tc} = (x_k^{tc}, \dot{x}_k^{tc}, \ddot{x}_k^{tc}, y_k^{tc}, \dot{y}_k^{tc}, \ddot{y}_k^{tc})^T$. Estimated position and velocity of the centre of a target group is used for standoff tracking guidance, which will be explained in the following section.

7.2 Multi-Target Tracking by a UAV

7.2.1 LVFG with variable standoff distance

This study applied a LVFG (Lyapunov Vector Field Guidance) for standoff group tracking, as introduced in Section 6.2.1.2. The LVFG uses the Lyapunov function:

$$V_l(x, y) = (r^2 - r_d^2)^2 \tag{7.1}$$

and the following desired dynamics in polar coordinates (hereafter the subscript k will be omitted for simple notation):

$$\begin{bmatrix} \dot{r} \\ r\dot{\theta} \end{bmatrix} = \frac{V_d}{r^2 + r_d^2} \begin{bmatrix} -(r^2 - r_d^2) \\ 2rr_d \end{bmatrix} \tag{7.2}$$

where $r = \sqrt{\delta x^2 + \delta y^2} = \sqrt{(x - x^{tc})^2 + (y - y^{tc})^2}$ is the distance of the UAV from the group enter. Herein (x^{tc}, y^{tc}) is the centre position of a target group estimated from the tracking filter as shown in Fig. 7.2, and V_d is a desired UAV speed. r_d is a desired standoff distance from the UAV to the centre of a target group which can be computed considering the FOV α_f of the UAV as:

$$r_d = \frac{d_{max} + d_m}{\sin\left(\frac{\alpha_f - \varepsilon_m}{2}\right)} \tag{7.3}$$

where d_{max} is the distance between group centre and the target furthest from the centre in the group, $d_m > 0$ a distance margin for d_{max}, and $\varepsilon_m > 0$ is an angle margin for the FOV of the UAV. Compared to the single target tracking where the target is located on the centre of sensor view, the effect of uncertainty or estimation error of target information becomes more crucial to keep all the targets in view of UAVs for the group tracking. Thus, this study exploits Mahalanobis distance concept [155, 156] to account for estimation error with relative uncertainties of the group centre as:

$$d_m = [\mathbf{z} - \hat{\mathbf{z}}]^T [P^{pos}_{k|k-1}]^{-1} [\mathbf{z} - \hat{\mathbf{z}}] \tag{7.4}$$

where $\hat{\mathbf{z}} = H\mathbf{x}^{tc}$ is the predicted measurement value and $P^{pos}_{k|k-1}$ is the position submatrix of the prediction covariance $P_{k|k-1}$. By using aforementioned standoff distance r_d, the UAV can keep all the target in the group within its FOV as shown in Fig. 7.2.

7.2.2 Convergence of the vector field to the standoff orbit

Since the standoff distance r_d varies according to the movement of the individual target in the group, the convergence of the vector field to the variable loiter circle (i.e. standoff orbit) is given as the following Lemma.

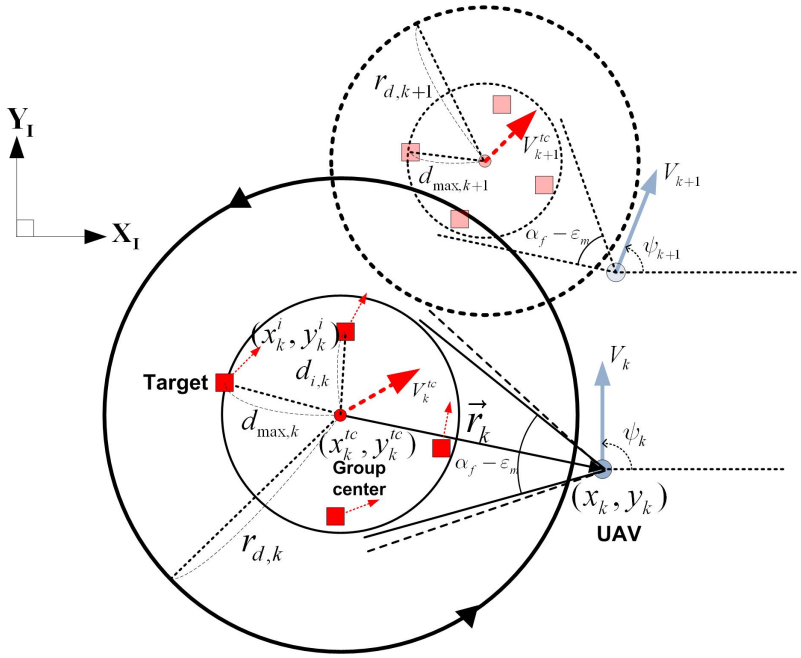


Figure 7.2: Geometric relation between UAV, ground target, and target group at time step k and $k+1$

Lemma 7.2.1. *If the threshold $\xi_{th} = (r\dot{r} - r_d\dot{r}_d)sign(r_d - r) \geq 0$, then the vector field as well as the UAV is globally stable to the loiter circle of the distance r_d and its rate $\dot{r}_d$.*

Proof. The proof will be represented with two cases depending on the radial distance of the vehicle.

case 1: $r < r_d$ (the UAV is inside the standoff orbit)

In this case, $\text{sign}(r_d - r) > 0$ and $\dot{r} > 0$ from Eq. (7.2). Let us consider the Lyapunov function $V_l(x, y) = (r^2 - r_d^2)^2$ as defined in Eq. (7.1) to check the convergence of r to r_d. This vector field produces a time rate of change of V_l:

$$\begin{aligned} \dot{V}_l(x, y) &= 4(r^2 - r_d^2)(r\dot{r} - r_d\dot{r}_d) \\ &= -\frac{4V_d r(r^2 - r_d^2)^2}{(r^2 + r_d^2)} - 4r_d(r^2 - r_d^2)\dot{r}_d. \end{aligned} \tag{7.5}$$

If $\xi_{th} = (r\dot{r} - r_d\dot{r}_d)\text{sign}(r_d - r) \geq 0$, then the fact that $(r\dot{r} - r_d\dot{r}_d)$ is less than zero (or equivalently, $\dot{r} > \frac{r_d}{r}\dot{r}_d$) makes $\dot{V}_l \leq 0$. Thus, r converges to the largest invariant set $r = r_d$ satisfying $\dot{V}_r = 0$ by LaSalles's invariance principle [145] except $r = 0$, which makes the vector field globally stable to the loiter circle. Since the UAV speed is set to track the constant vector field speed, the vehicle speed converges to

the vector field. Moreover, since the vector field is globally stable to the standoff orbit, so is the UAV.

case 2: $r \geq r_d$ (the UAV outside the standoff orbit)

In this case, $\text{sign}(r_d - r) \leq 0$ and $\dot{r} \leq 0$ from Eq. (7.2). Using the same Lyapunov function as above, it can be easily shown that if $\xi_{th} \geq 0$ (or equivalently, $\dot{r} \leq \frac{r_d}{r}\dot{r}_d$), then $\dot{V}_l \leq 0$. Thus, r converges to the largest invariant set $r = r_d$ from the outside the loiter circle. □

Remark 7.2.1. *The proof of Lemma 7.2.1 implies that if the sign of $\dot{r}_d$ is different from that of $\dot{r}$, then the vector field always converges to the loiter circle.*

For $\xi_{th} < 0$, since the vector field does not converge to the loiter circle from Lemma 7.2.1, and the following holds:

$$\begin{aligned} 0 &< \dot{r} < \frac{r_d}{r}\dot{r}_d, \text{ if } r < r_d \\ 0 &\geq \dot{r} \geq \frac{r_d}{r}\dot{r}_d, \text{ otherwise} \end{aligned} \tag{7.6}$$

, this study proposes two active measures to enhance the convergence of the vector field by increasing $|\dot{r}|$ such that $|\dot{r}| \geq |\frac{r_d}{r}\dot{r}_d|$. The first one is done by introducing k_l in the radial equation in Eq. (7.2) to adjust the convergence of the vector field as:

$$\dot{r}_{new} = -\frac{V_d}{k_l}\frac{r^2 - r_d^2}{r^2 + r_d^2} \tag{7.7}$$

where $0 < k_l \leq 1$ is a positive constant. By doing this, a rate of change of V_l in Eq. (7.5) also becomes faster as:

$$\dot{V}_l(x, y) = -\frac{4V_d r(r^2 - r_d^2)^2}{k_l(r^2 + r_d^2)} - 4r_d(r^2 - r_d^2)\dot{r}_d. \tag{7.8}$$

The second measure is to use the virtual standoff distance $r_{d,vir}$ in proportion to $\dot{r}_d$ as:

$$r_{d,vir} = r_d + k_{\dot{r}_d}\dot{r}_d \tag{7.9}$$

where $k_{\dot{r}_d}$ is a positive control gain. The basic idea of this is to exploit the approximated future standoff distance using the current change rate of r_d, rather than chasing the loiter circle behind it. It can be easily shown that substituting Eq. (7.9)

into Eq. (7.7) increases $|\dot{r}|$ for the UAV inside as well as outside the loiter circle. However, note that above two strategies do not guarantee the convergence of the vector field all the time, especially when r_d or $\dot{r}_d$ is big due to dispersion of the targets or a high speed vehicle in the target group. This leads to a condition for discarding target from the group which will be discussed in Section 7.3.2.

7.2.3 Vector field guidance command

The desired heading can be decided using the desired velocity components in Eq. (6.18) as:

$$\psi_d = \tan^{-1} \frac{\dot{y}_d}{\dot{x}_d} \tag{7.10}$$

The guidance command u_ω for turn rate is selected as the sum of proportional feedback and feedforward terms by differentiating Eq. (7.10) as:

$$u_\omega = -k_\omega(\psi - \psi_d) + \dot{\psi}_d \tag{7.11}$$

where

$$\dot{\psi}_d = 4V_d \frac{r_d r^2}{(r^2 + r_d^2)^2} - \frac{2r\dot{r}_d}{r^2 + r_d^2}. \tag{7.12}$$

$\dot{\psi}_d$ can be obtained by differentiating Eq. (7.10). As r approaches r_d, the left term of Eq. (7.12) increases monotonically, and magnitude of the right term also increases. Then, the guidance vector field will be feasible as long as the loiter circle pattern itself is feasible considering variable $\dot{r}_d$, which satisfies the following when $r = r_d$:

$$\dot{\psi}_d = \frac{V_d}{r_d} - \frac{\dot{r}_d}{r_d} < \omega_{max}. \tag{7.13}$$

Using above equation, a feasible standoff distance can be determined as:

$$r_d \geq \frac{V_d}{\omega_{max}} - \frac{\dot{r}_d}{\omega_{max}} = r_{d,min}. \tag{7.14}$$

Since $\dot{r}_d$ is bounded by the maximum speed of a ground vehicle, $r_{d,min}$ can be determined uniquely by the UAV kinematic constraints.

7.2.4 Taking a target group velocity into account

Since the velocity of the centre of each group can be estimated as explained in Section 7.1, the guidance vector can adjusted in order to take a target velocity into account. Let us define the following relation between the new desired velocity of the UAV $[\dot{x}_{dn}, \dot{y}_{dn}]^T$ and the velocity of the target group centre $[\dot{x}^{tc}, \dot{y}^{tc}]$ using a scale factor α_s and the desired x and y velocity components derived in Eq. (6.18) [14].

$$\begin{bmatrix} \dot{x}_{dn} \\ \dot{y}_{dn} \end{bmatrix} = \begin{bmatrix} \dot{x}^{tc} + \alpha_s \dot{x}_d \\ \dot{y}^{tc} + \alpha_s \dot{y}_d \end{bmatrix} \tag{7.15}$$

The condition so that the UAV flies with the desired speed V_d can be expressed by taking the norm of Eq. (7.15) as:

$$(\dot{x}_d^2 + \dot{y}_d^2)\alpha_s^2 + 2(\dot{x}_d \dot{x}^{tc} + \dot{y}_d \dot{y}^{tc})\alpha_s + (\dot{x}^{tc})^2 + (\dot{y}^{tc})^2 - V_d^2 = 0 \tag{7.16}$$

Above equation has one positive real solution for α_s only if the desired speed of the UAV is larger than the target speed. Substituting this solution into Eq. (7.15) yields the modified desired guidance vector of the UAV.

7.3 Coordinated Multi-Target Tracking by Multiple UAVs

This section proposes a multi-target group surveillance strategy by cooperating multiple UAVs with its benefits such as better estimation accuracy with sensor/data fusion and more robust tracking performance. Since multi-target tracking using multiple UAVs is typically NP-hard both in the number of sensing agents and targets [37], this study uses a two-step approach: (1) target clustering/resource allocation; (2) cooperative standoff group tracking with local replanning.

7.3.1 Target clustering and resource allocation

Since this study uses standoff tracking concept which is continuously orbiting around moving targets, one of sub-optimal approaches to partition the targets

would be treating geographically close targets as the same target group. This is done by K-means clustering algorithm to group objects based on attributes into pre-defined K number of groups [157]. As an efficient heuristic algorithm that is commonly employed, this converges quickly to the solution even to a large number of targets. Detailed analysis regarding computation complexity with the fixed number of groups and targets can be found in [158]. The grouping is done by minimising the sum of squares of distances between data and the corresponding cluster centroid as Algorithm 2 where the optimisation objective J is in the form:

Algorithm 2 K-means algorithm to cluster multiple targets

Input: K (number of clusters) and target position data $\{\mathbf{x}_{pos}^1, \cdots, \mathbf{x}_{pos}^m\}$

Randomly initialise K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^2$
while $\Delta J > \epsilon$ {J := optimisation objective [Eq. (7.17)]} **do**
  for $i = 1$ to m **do**
    Compute $c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $\mathbf{x}_{pos}^i$
  end for
  for $k = 1$ to K **do**
    Compute μ_k := average (mean) of target positions assigned to cluster k
  end for
end while

$$J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_2, \ldots, \mu_K\right) = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}_{pos}^i - \mu_{c^{(i)}}\| \tag{7.17}$$

where $\mu_{c^{(i)}}$ is index of cluster to which target point $\mathbf{x}_{pos}^i$ is currently assigned. This study considers the situation where either one or two UAVs are engaging the same target group, thus the number of clusters is determined by the number of UAVs N_u as $K = N_{tg} = \text{round}\left(\frac{N_u}{2}\right)$, where round$(\cdot)$ represents rounding inside element to the nearest integer.

After clustering, UAVs need to be assigned to the corresponding target group. The optimal assignment approach is used as the one that gathers the most information of targets using the Fisher information matrix (FIM). The FIM describes the amount of information a set of measurements contains about the state variable in terms of sensitivity of the estimation process [30]. Thus, maximising the FIM is more likely to improve the estimation performance and reduce uncertainty. In other words, initial assignment of UAVs that yields large values of some measure

of the FIM is expected to yield better estimation performance compared to those that gives lower values. The details of the FIM can be found in [30, 159]. Assuming that prior information is always ignored, the FIM for multiple UAVs to a single target is given as:

$$\begin{aligned} I^{FIM} &= \sum_{i=1}^{N_u} H_i^T R_i^{-1} H_i \\ &= \sum_{i=1}^{N_u} \begin{bmatrix} \frac{\cos^2\theta_i}{\sigma_r^2} + \frac{\sin^2\theta_i}{r_i^2\sigma_\theta^2} & \frac{\cos\theta_i \sin\theta_i}{\sigma_r^2} - \frac{\cos\theta_i \sin\theta_i}{r_i^2\sigma_\theta^2} \\ \frac{\cos\theta_i \sin\theta_i}{\sigma_r^2} - \frac{\cos\theta_i \sin\theta_i}{r_i^2\sigma_\theta^2} & \frac{\sin^2\theta_i}{\sigma_r^2} + \frac{\cos^2\theta_i}{r_i^2\sigma_\theta^2} \end{bmatrix} \end{aligned} \tag{7.18}$$

where θ_i represents the bearing angle of i-th UAV to the target. The determinant η_D=det(J) is used to measure the size of the FIM. Then, the assignment solution to maximise the FIM can be obtained by solving the following formulation:

$$\max\ J = \det\left(\sum_{i=1}^{N_s}\sum_{j=1}^{N_{tg}} I_{ij}^{FIM} x_{ij}\right) \tag{7.19}$$

$$\sum_{j=1}^{N_{tg}} x_{ij} \le 1,\ x_{ij} \in \{0,1\},\ \text{for } i = 1,\ldots,N_s \tag{7.20}$$

where $N_s = 2^{N_u} - 1$ is the number of possible combinations of N_u UAVs to observe the target group, I_{ij}^{FIM} represents the FIM of i-th UAV combination assigned to j-th target group. Eq. (7.20) represents that one target group is assigned to one UAV combination at most. Note that this optimisation process is performed only once at the initial stage.

7.3.2 Online local replanning

Once initial assignment of UAVs to target groups is done, online local replanning is followed either by handing over targets between groups or discarding target out of the group according to sensing range or the convergence of the vector field while UAVs are persistently following corresponding groups.

- **Target Hand-off:** By running K-means clustering algorithm in a recursive manner, target hand-off event between groups can be done inherently since clustering itself can regroup targets according to their proximity to the target

group and UAVs. To avoid frequent change of the group for the target on the boundary between two (passing/receiving) groups as well as make sure that the target passed to the receiving group is inside the FOV of UAVs of the receiving group, hand-off occurs for a certain period of time T_{hd}:

$$T_{hd} \geq \left[\frac{|r - r_d|}{|\dot{r}_{new} - \frac{r_d}{r}\dot{r}_d|} \right]_{t=t^i_{hd}} \tag{7.21}$$

where t^i_{hd} represents the time when the target is first requested for the hand-off by the clustering algorithm. For T_{hd}, the hand-off target will be included in both passing and receiving group. Until UAVs for the receiving group reach the desired standoff orbit keeping the hand-off target in their FOV, the UAV for the passing group sends the position of the hand-off target to the receiving group.

- **Target Discard:** If the standoff distance for the group tracking becomes larger than the sensing range (i.e. $r_d > r_{d,max}$), or radial velocity difference between the vector field and desired standoff orbit is larger than a certain value (i.e. $\xi_{th} < -\xi_d$ from Lemma 7.2.1), the target furthest from the centre in the group is removed from the group.

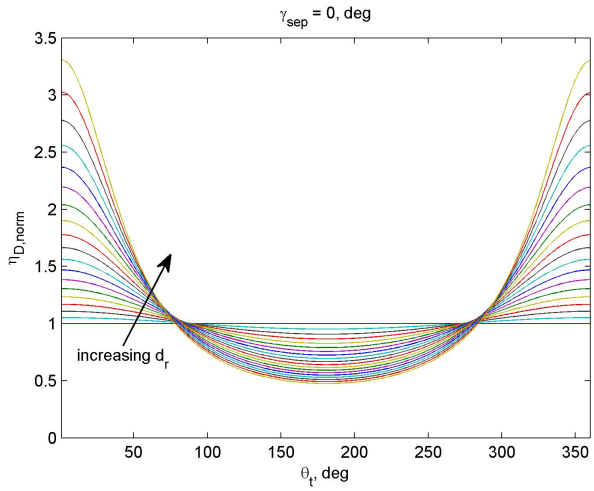
7.3.3 Sensitivity analysis to orbit coordination

In case that a pair of UAVs are involved for the same target group, the angular separation between UAVs is separately accomplished by controlling the speed of UAVs to obtain more accurate target information as [14]:

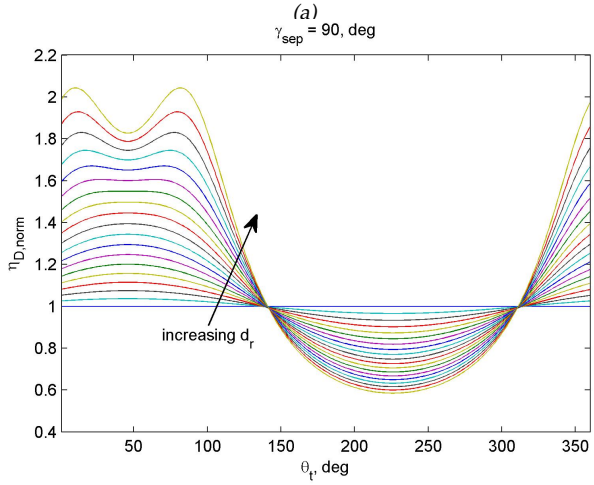
$$u_v = \pm k_v(\gamma - \theta_d)r_d + V_d \tag{7.22}$$

where k_v is a control gain, θ_i is the azimuth angle of the ith UAV relative to the group centre, $\gamma = \theta_2 - \theta_1$ is the angular phase separation of UAVs, and θ_d is a desired phase difference between the UAVs.

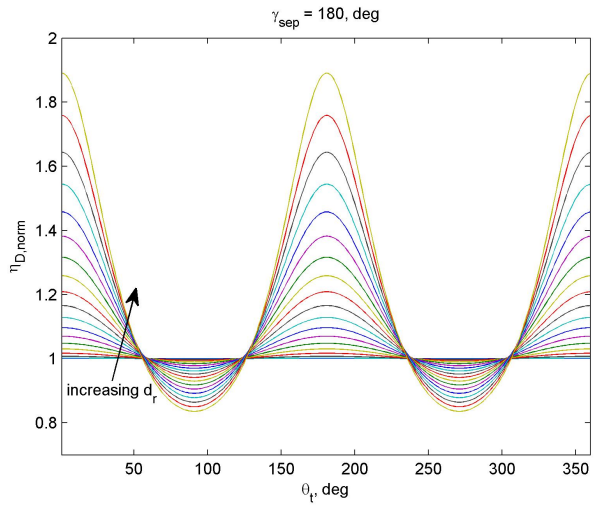
The desired phase difference θ_d can be determined differently depending on the objective of the mission such as estimation accuracy or visibility of an adversarial target. This study adopts the strategy which maximises information in the current measurements without considering previous information using the FIM



(a)



(b)



(c)

Figure 7.3: Nondimensional η_D w.r.t. target distance from the group centre (d_r) and specific angular separations ($\gamma_{sep} = 0, 90, 180°$) for $\sigma_{scale} = 1$

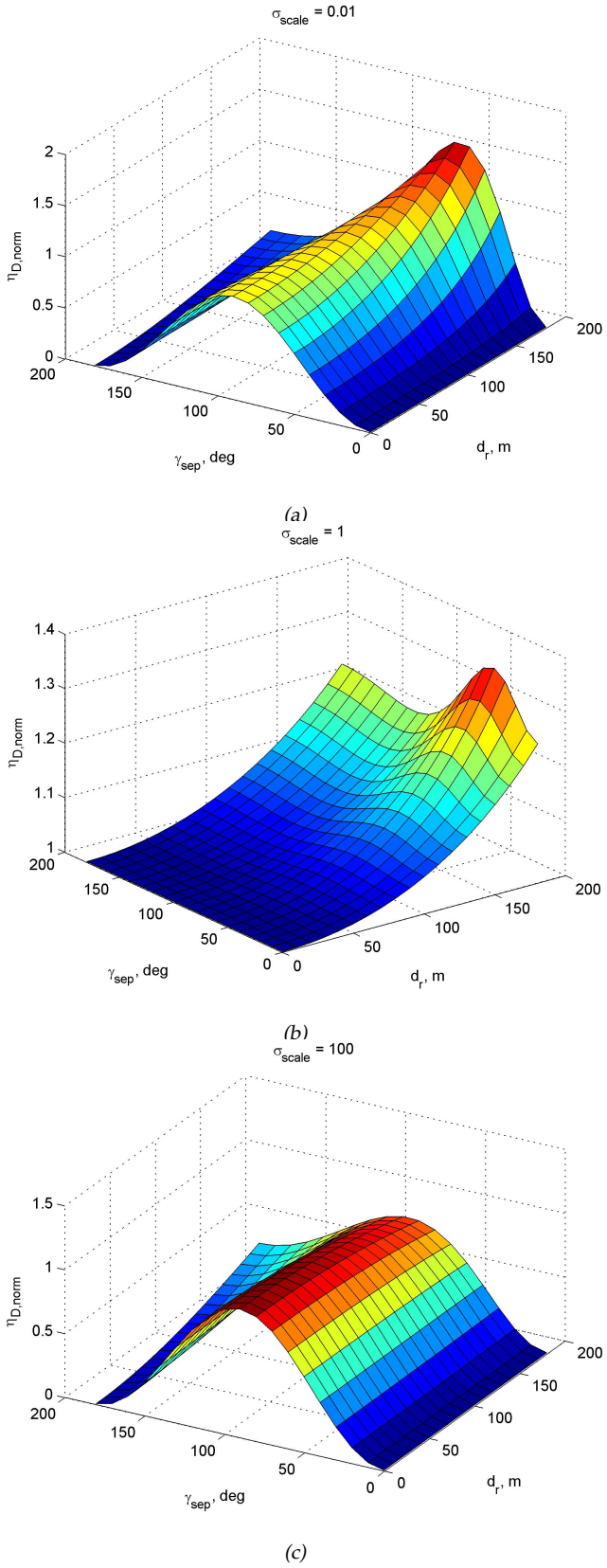


Figure 7.4: Nondimensional η_D (average for $0 \leq \theta_t < 360°$) w.r.t. target distance from the group centre (d_r) and angular separation (γ_{sep}) for different values of scaled noise standard deviation

as used in the previous section. The determinant of FIM from two UAVs for the same single target can be given using Eq. (7.18) and trigonometric identities as:

$$\begin{aligned} \eta_{D,pair} &= \det\left(\sum_{i=1}^{2} H_i^T R_i^{-1} H_i\right) \\ &= \frac{1+\cos^2\gamma}{\sigma_r^2\sigma_\theta^2}\left(\frac{1}{r_1^2}+\frac{1}{r_2^2}\right)+\left(\frac{\sin^2\gamma}{\sigma_r^4}+\frac{\sin^2\gamma}{\sigma_\theta^4 r_1^2 r_2^2}\right). \end{aligned} \tag{7.23}$$

For the single target tracking, the optimal value of γ which maximises η_D can be analytically obtained from Eq. (7.23). However, in this study, since the targets are dispersed around the group centre, it is difficult for a pair of UAVs loitering the same target group to determine one specific optimal γ. To check a tendency of η_D depending on target position specified with the range $0 \le d_r \le 180$ m and angle $0 \le \theta_t < 360°$ from the group centre, numerical analysis is performed for different angular separation γ_{sep} with fixed $r_d = 400$ m as illustrated in Fig. 7.5. For the analysis, the standard deviation of measurement noise is generalised with $\sigma_{scale} = \frac{r_d\sigma_\theta}{\sigma_r}$ such that small scale represents sensor characteristic close to bearing-only and large value for pure ranging.

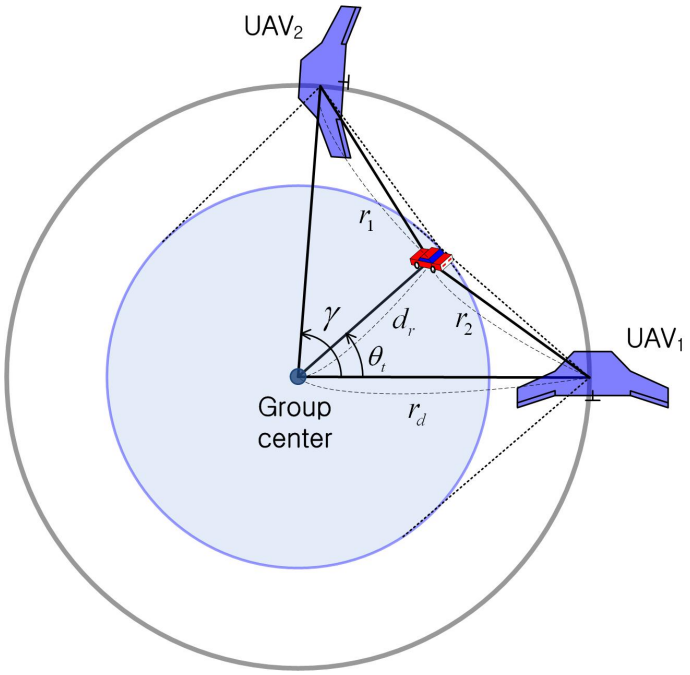


Figure 7.5: Geometric relation between UAVs and a ground target

Firstly, Fig. 7.3 shows the example of $\eta_{D,norm}$ (scaled by $\eta_{D,0}$ with $d_r = 0$, $\theta_t = 0°$, and $\gamma_{sep} = 90°$) for specific angular separations ($\gamma_{sep} = 0, 90, 180°$) for $\sigma_{scale} = 1$.

Figure 7.4 shows $\eta_{D,norm}$ (averaged for all θ_t) for different σ_{scale} with respect to the target distance d_r and angular separation γ_{sep}. For $\sigma_{scale} = 0.01$ in Fig. 7.4(a), both the optimality criterion and the optimal value of γ change as a function of the range d_r since this configuration is close to bearing-only measurement as explained above. For $\sigma_{scale} = 100$ (or, pure ranging), the optimality criterion changes as γ_{sep} changes and optimal value remains around $\pi/2$ independent of d_r. Lastly, in case of the sensor with $\sigma_{scale} = 1$ (i.e. range and bearing sensor such as a GMTI), $\eta_{D,norm}$ is only a function of range for small d_r (< 100 m), however, angular separation has some effect on it as d_r gets larger. Based on this analysis, an online algorithm for optimal γ for group target tracking could be developed, however, this remains as future work.

7.4 Numerical Simulations

This section carries out numerical simulations using the proposed standoff group tracking algorithm for moving ground targets using four UAVs to show the feasibility and benefits of the proposed approach. The true target trajectories (six randomly moving by target model and one manoeuvring) are used to generate GMTI measurements at 2 Hz mixing with white gaussian noise. The parameters used for the simulation are shown in Table 7.1.

Figure 7.7 shows the absolute trajectories of standoff tracking of seven ground target (six randomly moving and one manoeuvering) with four UAVs. Firstly, targets are clustered into two groups, and UAVs are assigned to the appropriate group using the proposed assignment algorithm as shown in Fig. 7.7(a). Note that data association regarding which measurement comes from which target is assumed to be solved as mentioned in Section 7.1 in this study. However, since our approach to track multiple targets is to exploit the centre of the group and furthest target information from the centre only, even some false data association at the beginning in a cluttered situation as shown in Fig. 7.7(a) would not affect the guidance performance in terms of keeping all targets in the standoff orbit. At around 35 second, since target hand-off event triggered, the target (moving towards north-east direction) is included in both target groups until UAVs of the receiving group (group 2) reach the desired standoff orbit for T_{hd} seconds as shown in Fig. 7.7(b) and Fig. 7.6(a). Figure 7.7(c) shows the situation after target

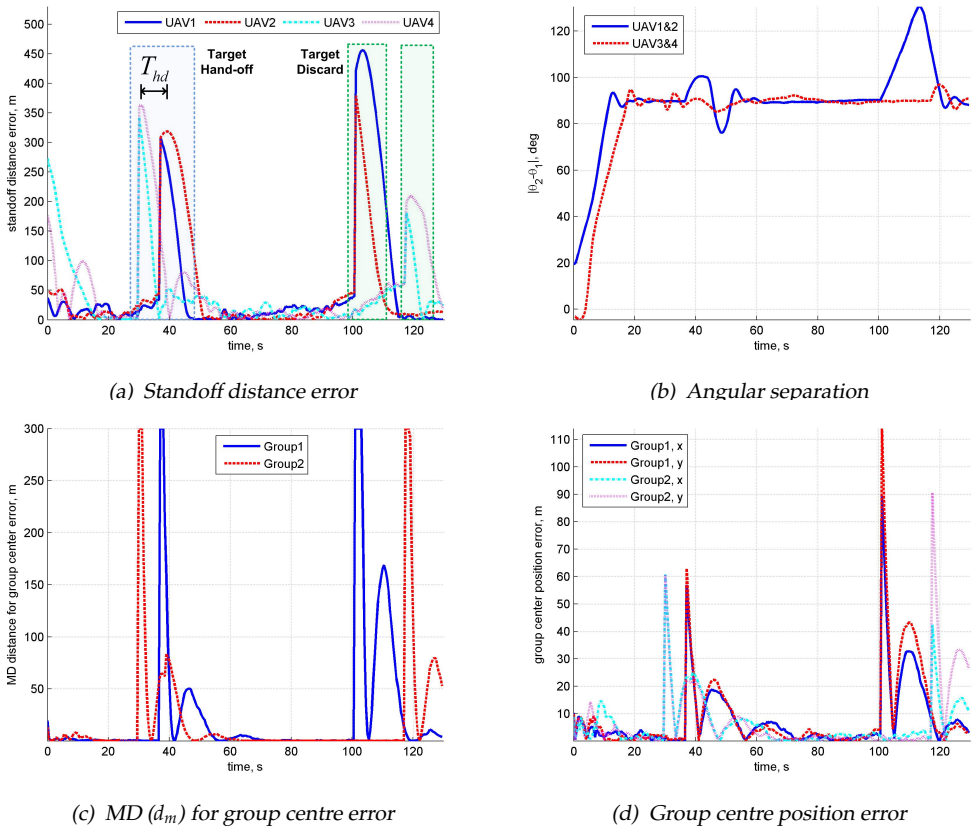
Table 7.1: Simulation parameters

Parameter	*Value*	*Unit*
T_s	0.5	sec
α	0.1	N/A
(α_f, ϵ_m)	(70,10)	deg
$(\sigma_r, \sigma_\theta)$	(10, 3)	(m,deg)
θ_d	90	deg
v_0	40	m/s
$(r_{d,min}, r_{d,max})$	(250,700)	m
v_{max}	15	m/s
ω_{max}	0.3	rad/s
τ_v, τ_ω	1/3	sec
$(k_v, k_w, k_l, k_{\dot{r}_d})$	(0.5,2,0.3,1.5)	N/A

hand-off (from group 1 to group 2) process is finished. As targets in the group get dispersed widely, the furthest target from the centre is removed from the group depending on the sensing range or convergent limit as introduced in Section 7.3.2. Mahalanobis distance in Fig. 7.6(c) is used to account for estimation error of group centre position when the group centre changes abruptly due to target hand-off or discard, and shows the same tendency as centre position estimation error (i.e. $|\mathbf{x}^{tc}_{true} - \mathbf{x}^{tc}|$) as shown in Fig. 7.6(d). Finally, Figure 7.6~7.7 show successful cooperative standoff group tracking in terms of standoff distance error and desired angular separation keeping while placing all targets of interest inside the FOV of the UAV at all times in a dynamic environment.

7.5 Summary

This chapter has presented the coordinated standoff tracking of groups of moving targets using multiple UAVs. Based on the vector field guidance approach, an active sensing/guidance algorithm was developed to maximise information of the targets and keep all targets inside the view of multiple UAVs considering physical and sensing constraints. For multiple group target surveillance by multiple UAVs,



(a) Standoff distance error

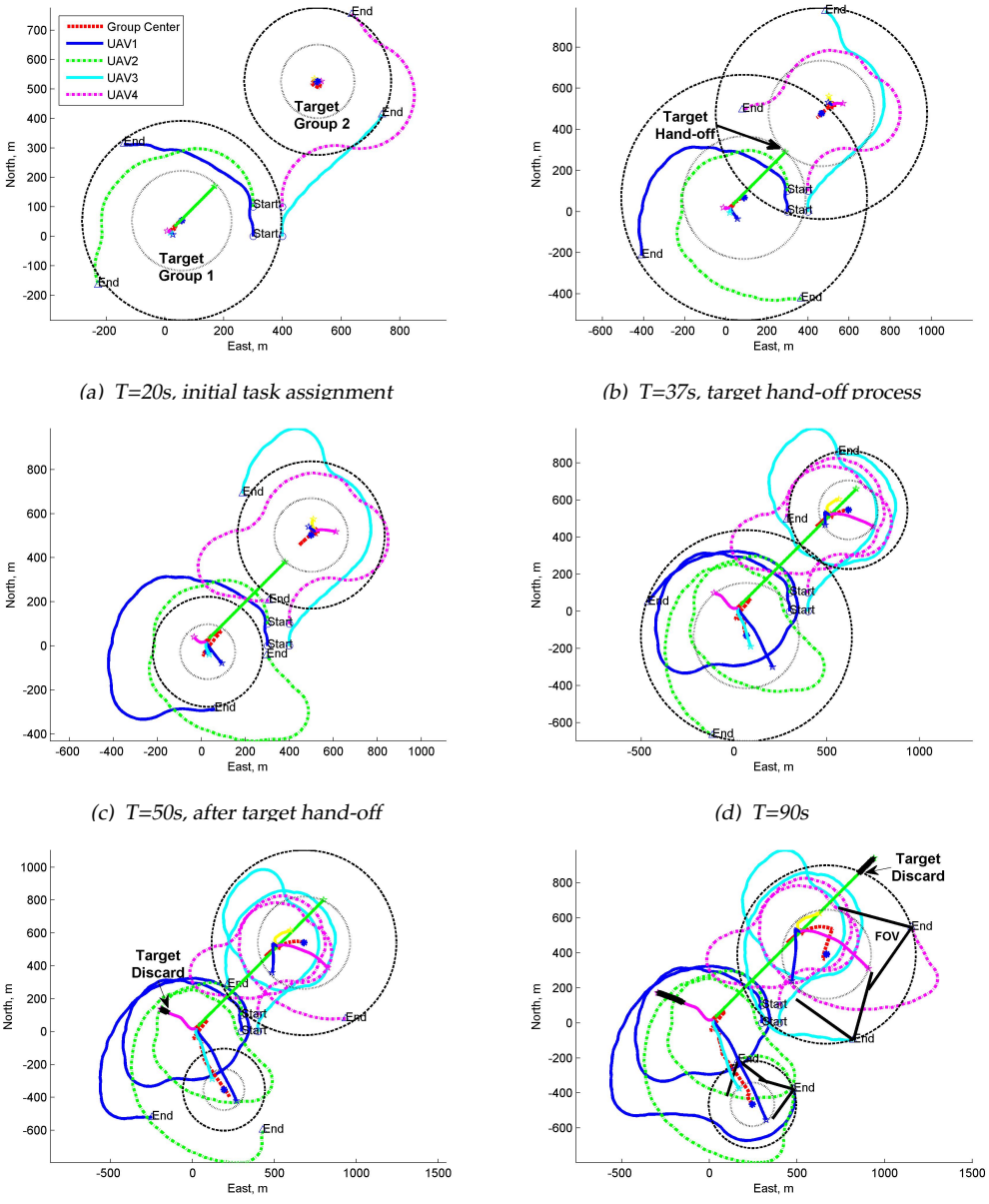
(b) Angular separation

(c) MD (d_m) for group centre error

(d) Group centre position error

Figure 7.6: Standoff tracking simulation results of seven ground targets (six randomly moving and one manoeuvring) with four UAVs

a two-phase approach was proposed consisting of target clustering/assignment and cooperative standoff tracking with online local replanning including target hand-off and discard from the group. Localisation sensitivity to the group of targets was also investigated for different angular separations between UAVs and sensing configurations as a basis of future optimal separation scheme. Numerical simulation showed successful standoff group tracking as well as local replanning while keeping all targets of interest within the FOV of the UAV at all times in a dynamic environment.



(a) T=20s, initial task assignment

(b) T=37s, target hand-off process

(c) T=50s, after target hand-off

(d) T=90s

(e) T=110s, target discard in group 1

(f) T=130s, target discard in group 2

Figure 7.7: Absolute trajectories of standoff tracking of seven ground targets (six randomly moving and one manoeuvring) with four UAVs

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis has investigated the use of a team of small and low-cost UAVs for autonomous aerial surveillance, which aims to identify and continuously track suspicious vehicles and disguised threats in the ground traffic, and then notify human operators about potentially dangerous vehicles in the area of interest. A group of UAVs become a swarm of airborne sensor platforms, and tasks and routes of each UAV need to be planned in order to cooperatively achieve a surveillance mission. Directions to search or monitor particular areas and to track potential targets obtained by the behaviour analysis are typical instructions. To satisfy required surveillance capabilities under a deluge of data and information from complex scenes, multi-sensor management and information fusion are extensively addressed allowing mobile sensor platforms to intelligently obtain and process target information for anomalous behaviour monitoring.

Among various techniques for autonomous surveillance, this thesis developed and integrated several necessary components, exploiting geographical, spatial and temporal information associated with operational environments. Firstly, practical approaches were presented for a road-network search strategy by which a team of UAVs visit every road in the map of interest. The MILP optimization and the nearest insertion algorithm along with the auction negotiation were proposed. To realistically accommodate the manoeuvreing constraints of UAVs, the Dubins path planning was used. The performance of the proposed algorithm was investigated via a Monte Carlo simulation by analyzing the effects of different map sizes, path planning methods and the number of UAVs. Based on these results, the efficient UAV team size and path planning method were suggested for the route planning, and this framework can be applied to a variety of autonomous search missions in the initial phase of mission planning.

This thesis also presented an integrated behaviour recognition methodology for ground vehicles using UAVs, to identify suspicious or abnormal behaviour

combining the following techniques: a target tracking filter, sensor fusion, trajectory classification, behaviour detection and recognition. Numerical simulation results using real VMTI measurements demonstrated the feasibility of the proposed approach. Furthermore, a synthetic military scenario and realistic car trajectory data from the off-the-shelf traffic S-Paramics simulation program were exploited to extend our approach to more general cases, by introducing a fuzzy rule-based decision making algorithm, as well as a Gaussian regression process. The proposed methodology allows suspicious behaviour to be automatically identified to enable increased situational awareness in a much faster and efficient manner and hence reduce the operator workload significantly.

Lastly, this thesis presented the coordinated standoff tracking of groups of moving targets using multiple UAVs, some of which is in conjunction with a road-map assisted filtering and sensor fusion techniques. Four tracking algorithms were developed for a single ground target, and their properties and performance were compared with one another. This study particularly investigated the effect of improved estimation accuracy on the tracking guidance performance. In addition, based on the vector field guidance approach, an active guidance algorithm was developed to maximise information of the multiple targets and keep all of them inside the view of UAVs considering physical and sensing constraints. For multiple group target surveillance by multiple UAVs, a two-phase approach was proposed consisting of target clustering/assignment and cooperative standoff tracking with online local replanning including target hand-off and discard from the group. Numerical simulation showed successful standoff tracking as well as local replanning while keeping all targets within the FOV of the UAV sensor at all times in a dynamic environment.

From this thesis, it has been identified that following aspects need to be carefully considered to realise autonomous surveillance using multiple UAVs: i) how many UAVs/sensors would be enough to perform a mission in terms of efficiency, estimation accuracy and guidance performance, ii) information gathered by UAVs only is enough, or domain knowledge (local context and past experience) might be additionally required, iii) communication structure between UAVs, and iv) computation time. The proposed autonomous surveillance system using multiple UAVs can greatly increase the amount of area continuously monitored, while reducing the number of human operators and their workload required to anal-

yse surveillance data and respond to identified targets. Application examples include, monitoring urban/rural area, operations such as peace keeping and force protection, detecting unknown intent of terrorists and insurgents, and providing a protective surveillance network around military facilities. Moreover, this research can be easily extended to the maritime and civilian domain, for enhancing situational awareness of traffic movements both on land and at sea.

8.2 Future Work

Future work will be performed to complete the autonomous surveillance and target tracking mission using cooperative UAVs as follows. Firstly, although the proposed road-network route planning algorithm can be extended for the unstructured area search by generating artificial road patterns, more dynamic and flexible approaches need to be designed to enhance the performance of target search for an uncertain environment. One possible approach is first to build the probability/information map of target existence given the expected position of target [160, 161]. The probability is computed in proportion to the accuracy and importance of the available sensor data. Next, the point or place that has probability more than the specific threshold value is determined as expected target points, and then final waypoints or paths are generated by considering the terrain and obstacles. In addition, multi-objective route planning can be adopted for this problem considering time taken, likelihood of detection and fuel consumed with real-world data sources [162, 163].

Another future work will be a persistent surveillance. The task of persistent surveillance is somewhat different from typical coverage or search problems in a sense that the area of interest needs to be continuously searched while minmising the time between visitations to the same region to deal with pop-up or hidden targets [76]. The pop-up or hidden targets might appear or disappear at frequent and random time intervals, which can cause significant challenges in designing and implementing a cooperative search strategy. Although the concept of the presented search route planning algorithm can still be applied, a cost function (which was to minimise a flight time covering all the roads or area in this thesis) is required to be adjusted according to the objective of the surveillance mission.

Regarding airborne behaviour monitoring, additional relevant aspects of behaviour will be considered as decision inputs. An example is the cultural background related to driving habits in the region where this system is to be deployed. The performance and robustness of the proposed scheme will also be investigated using both Monte Carlo simulation as well as real GMTI sensor data for different types of cars. Application of an adaptive fuzzy system based on machine learning [164] would be of interest for more flexible and robust human-environment interactions.

Since this thesis addressed only the initial proof of concept for coordinated standoff tracking of moving targets, various implementation issues will be tackled as future work such as: the effect of imperfect communication amongst UAVs; measurement data association for the target estimation in conjunction with target group clustering; and optimal angular spacing strategies in consideration of line-of-sight block by obstacles in an urban environment.

In this thesis, the mission control station is assumed to have dedicated operators reviewing and approving the decision and ensuring the right information get delivered to the right UAV, using extensive automation tools to facilitate the task of managing the resources. However, for full autonomy, autonomous decision making algorithm needs to take the role of human operators. A team of UAVs should be able to cooperate and decide when, which and how many UAVs will be assigned to a set of given tasks. In particular, endowing an airborne sensor swarm with decision making autonomy requires modelling architectures that allow the sensor swarm to organise them in groups and to manage its mobility, reconfiguration and accommodate failure of the individual sensor platforms. These decision making aspects, capable of performing in real-time, dealing with an imperfect communication, and accommodating unknown dynamic environments, will be considered in future research.

References

References

[1] http://www.as.northropgrumman.com/products/hart/index.html.

[2] http://www.roke.co.uk.

[3] http://maps.google.co.uk/maps.

[4] T. Samad, J.S. Bay, and D. Godbole. Network-centric systems for military operations in urban terrain: The role of UAVs. *Proceedings of the IEEE*, 95(1):92–107, 2007.

[5] Office of the Secretary. Unmanned aircraft systems roadmap 2005-2030. Technical report, U.S. Department of Defense, 2005.

[6] P.M. Miller. Mini, micro, and swarming unmanned aerial vehicles: A baseline study. Technical report, Federal Research Division, Library of Congress, Washington, D.C., November 2006.

[7] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J.K. Hedrick. An overview of emerging results in cooperative UAV control. In *43rd IEEE Conference on Decision and Control*, pages 602–607, Atlantis, Bahamas, December 2004.

[8] H. Chung, S. Oh, D.H. Shim, and S.S. Sastry. Toward robotic sensor webs: Algorithms, systems, and experiments. *Proceedings of the IEEE*, 99(9):1562–1586, 2011.

[9] R.W. Beard, T.W. McLain, D.B. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *Proceedings of the IEEE*, 94(7):1306–1324, 2006.

[10] B.A. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Zbkowski. Contaminant cloud boundary monitoring using network of UAV sensors. *IEEE Sensors Journal*, 8(10):1681–1692, 2008.

[11] A.R. Girard, A.S. Howell, and J.K. Hedrick. Border patrol and surveillance missions using multiple unmanned air vehicles. In *43rd IEEE Conference on*

Decision and Control, pages 620–625, Paradise Island, Bahamas, December 2004.

[12] M.A. Goodrich, B.S. Morse, D. Gerhardt, J.L. Cooper, M. Quigley, J.A. Adams, and C. Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, 25:89–110, 2008.

[13] D. Murphy and J. Cycon. Applications for mini VTOL UAV for law enforcement. In *Information and Training Technologies for law enforcement*, Boston, MA, November 1998.

[14] E.W. Frew, D.A. Lawrence, and S. Morris. Coordinated standoff tracking of moving targets using Lyapunov guidance vector fields. *Journal of Guidance, Control, and Dynamics*, 31(2):290–306, 2008.

[15] D.W. Casbeer, D.B. Kingston, R.W. Beard, and T.W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[16] L. David. *Surveillance Studies: An Overview*. Cambridge: Polity Press, 2007.

[17] A. Puri. A survey of unmanned aerial vehicles. Technical report, Department of Computer Science and Engineering , University of South Florida, 2004.

[18] C.C. Haddal and J. Gertler. Homeland security: Unmanned aerial vehicles and border surveillance. Technical report, Congressional Research Service, 2010.

[19] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviours. *IEEE Transactions on Systems, Man, and Cybernatics-Part C: Applications and Reviews*, 34(3):334–352, 2004.

[20] G. Mathews, A. Waldock, and M. Paraskevaides. Toward a decentralised sensor management system for target acquisition and track. In *Systems Engineering for Autonomous Systems (SEAS) - Defence Technology Centre (DTC) Conference*, Edinburgh, UK, July 2010.

[21] M. Ulmke and W. Koch. Road-map assisted ground moving target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42:1264–1274, 2006.

[22] W. Koch and R. Klemm. Ground target tracking with STAP radar. *IEE Proceedings-Radar, Sonar and Navigation*, 148:173–185, 2001.

[23] M. Schreuder, S.P. Hoogendoorn, H.J. Van Zulyen, B.Gorte, and G. Vosselman. Traffic data collection from aerial imagery. In *IEEE Proceedings of Intelligent Transportation Systems*, volume 1, pages 779–784, 2003.

[24] S. Ossen. *Longitudinal driving behaviour: theory and empirics*. PhD thesis, Faculty of Civil Engineering and Geosciences, Transport and Planning Department, Technical University of Delft, 2008.

[25] N. Xiong and P. Svensson. Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3:163–186, 2002.

[26] C.G. Looney and L.R. Liang. Cognitive situation and threat assessments of ground battlespaces. *Information Fusion*, 4:297–308, 2003.

[27] K.B. Ariyur and K.O. Fregene. Autonomous tracking of a ground vehicle by a uav. In *American Control Conference*, Westin Seattle Hotel, Seattle, Washington, USA, 2008.

[28] J. Kim and Y. Kim. Moving ground target tracking in dense obstacle areas using UAVs. In *IFAC World Congress*, COEX, South Korea, 2008.

[29] R.A. Wise and R.T. Rysdyk. UAV coordination for autonomous target tracking. *AIAA Guidance, Navigation and Control Conference, Keystone, Colorado, USA*, 2006.

[30] E.W. Frew. Sensitivity of cooperative target geolocalization to orbit coordination. *Journal of Guidance, Control, and Dynamics*, 31(4):1028–1040, 2008.

[31] P. Svensson. Technical survey and forecast for information fusion. In *Military Data and Information Fusion, RTO meeting proceedings MP-IST-040*, Prague, Czech Republic, October 2003 2003.

[32] T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, and I. Kadar. Ground target tracking with variable structure IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):26–46, 2000.

[33] Z. Tang and U. Ozguner. Sensor fusion for target tracking maintenance with multiple UAVs based on bayesian filtering method and hospitability map. In *IEEE Conference on Decision and Control*, pages 19–24, Hawaii, USA, 2003.

[34] J.G. Herrero, J.A.B. Portas, and J.R.C. Corredera. Use of map information for tracking targets on airport surface. *IEEE Transactions on Aerospace and Electronic Systems*, 39(2):675–693, 2003.

[35] L. Parker. Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, 12(3):231–255, 2002.

[36] B. Jung and G.S. Sukhatme. A region-based approach for cooperative multi-target tracking in a structured environment. In *IEEE/RSJ International conference on Intelligent Robots and Systems*, pages 2764–2769, 2002.

[37] Z. Tang and U. Ozguner. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21:898–908, 2005.

[38] J. Ousingsawat and M.E. Campbell. Multiple vehicle team tasking for cooperative estimation. In *American Control Conference*, 2004.

[39] T. Shima and S. Rasmussen. *UAV Cooperative Decision and Control: Challenges and Practical Approaches*. SIAM, Philadelphia, 2010.

[40] A. Waldock and D. Nicholson. A framework for cooperative control applied to a distributed sensor network. *The Computer Journal*, 54(3):471–481, 2011.

[41] Y. Jin, Y. Liao, A. Minai, and M. Polycarpou. Balancing search and target response in cooperative unmanned aerial vehicle teams. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 36(3):571–587, 2006.

[42] P. Dasgupta. A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(3):549–563, 2008.

[43] A.E. Gil, K.M. Passino, and A. Sparks. Cooperative scheduling of tasks for networked uninhabited autonomous vehicles. In *IEEE Conference on Decision and Control*, pages 522–527, Hawaii, USA, December 2003.

[44] L.E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[45] S.M. Malaek and A.R. Kosari. Novel minimum time trajectory planning in terrain following flights. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):2–12, 2007.

[46] P. Chandler, S. Rasmussen, and M. Pachter. UAV cooperative path planning. In *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, Aug. 14-17 2000. AIAA-2000-4370.

[47] K. Yang S. Sukkarieh. 3D smooth path planning for a UAV in cluttered natural environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, Sep. 22-26 2008.

[48] U. Zengin and A. Dogan. Probabilistic trajectory planning for UAVs in dynamic environments. In *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, Illinois, Sep. 20-23 2004. AIAA 2004-6528.

[49] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference*, pages 1936–1941, Anchorage, AK, May. 8-10 2002.

[50] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, and A.N. Kostaras. Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 33(6):898–912, 2003.

[51] E. Edison and T. Shima. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 38(1):340–356, 2011.

[52] J.A. Reeds and R.A. Shepp. Optimal paths for a car that goes both forward and backward. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[53] J. Boissonnat, A. Cï£¡rï£¡zo, and J. Leblond. Shortest paths of bounded curvature in the plane. *Journal of Intelligent and Robotic System*, 11:5–20, 1994.

[54] A.M. Shkel and V.J. Lumelsky. Classification of the Dubins set. *Robotics and Autonomous Systems*, 34:179–202, 2001.

[55] A. Tsourdos, B. White, and M. Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles*. John Wiley & Sons, 2010.

[56] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[57] C. Tomlin, I. Mitchell, and R. Ghosh. Safety verification of conflict resolution manoeuvres. *IEEE Transactions on Intelligent Transportation System*, 2(2):110–120, 2001.

[58] M. Shanmugavel, A. Tsourdos, B. A. White, and R. Zbikowski. Differential geometric path planning of multiple UAVs. *Journal of Dynamic Systems, Measurement, and Control*, 129:620–632, 2007.

[59] M. Shanmugavel, A. Tsourdos, B.A. White, and R. Zbikowski. Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs. *Control Engineering Practice*, 18(9):1084–1092, 2010.

[60] J.L. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, 2003.

[61] J. Bellingham, M. Tillerson, A. Richards, and J. How. *Multi-Task Allocation and Path Planning for Cooperating UAVs, Cooperative Control: Models, Applications and Algorithms*. Kluwer Academic Publishers, 2003.

[62] P.R. Chandler, M. Pachter, D. Swaroop, J.M. Hwlett, S. Rasmussen, C. Schumacher, and K. Nygard. Compexity in UAV cooperation control. In *American Control Conference*, pages 1831–1836, American Control Conference, Anchorage, AK, 2002.

[63] J. Ryan, T. Bailey, J. Moore, and W. Carlton. Reactive tabu search in unmanned aerial reconnaissance simulations. In *30th Conference on Winter Simulation*, pages 873–880, Washington, DC, 1998.

[64] H. Choset. Coverage for robotics-a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.

[65] W.L. Pearna and W.C. Chiub. Approximate solutions for the maximum benefit chinese postman problem. *International Journal of Systems Science*, 36:815–822, 2005.

[66] N. Perrier, A. Langevin, and J.F. Campbell. A survey of models and algorithms for winter road maintenance. *Computers and Operational Research, Part IV: Vehicle Routing and Fleet Sizing for Plowing and Snow Disposal*, 34:258–294, 2007.

[67] K. Easton and J. Burdick. A coverage algorithm for multi-robot boundary inspection. In *IEEE International Conference on Robotics and Automation*, 2005.

[68] B. Alspach. Searching and sweeping graphs: a brief survey. *Matematiche (Catania)*, 59:5–37, 2006.

[69] K. Salva, E. Frazzoli, and F. Bullo. Traveling salesperson problems for the Dubins vehicle. *IEEE Trans. on Automatic Control*, 53(6):1378–1391, 2008.

[70] J. Le Ny, E. Feron, and E. Frazzoli. On the Dubins travelling salesman problem. *IEEE Transactions on Automatic Control*, 57:265–270, 2012.

[71] S. Rathinam, R. Sengupta, and S. Darbha. A resource allocation algorithm for multi-vehicle systems with non holonomic constraints. *IEEE Transactions on Automation Sciences and Engineering*, 4:98–104, 2007.

[72] A. Ahmadzadeh, G. Buchman, P. Cheng, A. Jadbabaie, J. Keller, V. Kumar, and G. Pappas. Cooperative control of UAVs for search and coverage. In *Conference on Unmanned Systems*, pages 1–14, 2006.

[73] H. Oh, H.S. Shin, A. Tsourdos, B.A. White, and P. Silson. Coordinated road network search for multiple UAVs using Dubins path. In *1st CEAS Specialist Conference on Guidance, Navigation and Control*, Munich, Germany, 2011.

[74] D. Ahr. *Contributions to Multiple Postmen Problems*. PhD thesis, Heidelberg University, 2004.

[75] I. Maza and A. Ollero. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. *Distributed Autonomous Robotic Systems 6*, 5:221–230, 2007.

[76] N. Nigam and I. Kroo. Persistent surveillance using multiple unmanned air vehicles. In *Aerospace Conference, 2008 IEEE*, Big Sky, MT, 2008.

[77] M. Hifi, M. Michrafy, and A. Sbihi. A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications*, 33:271–285, 2006.

[78] T. Ralphs, L. Ladanyi, M. Guzelsoy, and A. Mahajan. Symphony 5.2.4. `http://projects.coin-or.org/SYMPHONY`.

[79] D.B. Kingston and C.J. Schumacher. Time-dependent cooperative assignment. In *American Control Conference*, Portland, OR, 2005.

[80] M. Alighanbari. Task assignment algorithms for teams of UAVs in dynamic environments. Master's thesis, Massachusetts Institute of Technology, 2004.

[81] D. J. Rosenkrantz, R.E. Stearns, and P.M. Lewis. An analysis of several heuristics for the traveling salesman problem. *Fundamental Problems in Computing*, 1:45–69, 2009.

[82] D.P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *COMPUTATIONAL OPTIMIZATION AND APPLICATIONS*, 1(1):7–66, 1992.

[83] D.P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[84] W. Koch, J. Koller, and M. Ulmke. Ground target tracking and road map extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61:197–208, 2006.

[85] S. Kanchanavally, R. Ordonez, and J. Layne. Mobile target tracking by networked uninhabited autonomous vehicles via hospitability maps. In *American Control Conference*, Boston, USA, 2004.

[86] Z.M Kassas and U. Ozguner. A nonlinear filter coupled with hospitability and synthetic inclination maps for in-surveillance and out-of-surveillance tracking. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Application and Reviews*, 40(1):87–97, 2010.

[87] D. Simon and T.L. Chia. Kalman filtering with state equality constraints. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

[88] M. Zhang, S. Knedik, and O. Loffeld. An adaptive road-constrained imm estimator for ground target tracking in GSM networks. *International Conference on Information Fusion*, 2007.

[89] M. Tahk and J.L. Speyer. Target tracking problems subject to kinematic constraints. *IEEE Transactions on Automatic Control*, 35(3):324–326, 1990.

[90] M.S. Arulampalam, N. Gordon, M. Orton, and B. Ristic. A variable structure multiple model particle filter for gmti tracking. *International Conference on Information Fusion*, 2002.

[91] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[92] B. Ristic, S. Arulampalm, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[93] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 6(4):473 –483, 1970.

[94] SIAS Limited. S-paramics software. http://www.sias.com, Jan 2011.

[95] K. Mehrotra and P.R. Mahapatra. A jerk model for tracking highly maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1094–1105, 1997.

[96] S. Kim, H. Oh, and A. Tsourdos. Nonlinear model predictive coordinated standoff tracking of moving ground vehicle. *Journal of Guidance, Control and Dynamics*, 36(2):557–566, 2013.

[97] D.B. Barber, J.D. Redding, T.W. Mclain, R.W. Beard, and C.N. Taylor. Vision-based target geo-location using a fixed-wing miniature air vehicle. *Journal of Intelligent and Robotic Systems*, 47:361–382, 2006.

[98] F.L. Lewis. *Applied optimal control and estimation*. Prentice Hall Englewood Cliffs, NJ, 1992.

[99] A.G.O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press LLC, Boca Raton, Florida, 1998.

[100] S.J. Julier and J.K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American control conference*, pages 2369–2373, Albuquerque, New Mexico, 1997.

[101] D. Simon. Kalman filtering with state constraints: A survey of linear and nonlinear algorithms. *IET Control Theory and Applications*, 4:1303–1318, 2010.

[102] F.L. Lewis. *Optimal Estimation: With an Introduction to Stochastic Control Theory*. John Wiley & Sons, 1986.

[103] M.S.Grewal and A.P. Andrews. *Kalman Filtering: Theory and Pratice Using MATLAB*. John Wiley and Sons, Inc., 2008.

[104] J.Z. Sasiadek. Sensor fusion. *Annual Reviews in Control*, 26(2):203–228, 2002.

[105] P.O. Arambel, C. Rago, and R.K. Mehra. Covariance intersection algorithm for distributed spacecraft state estimation. In *American Control Conference*, 2001.

[106] M.E. Liggins, D.L. Hall, and J. Linas. *Handbook of Multisensor Data Fusion: Theory and Practice, second edition*. CRC Press, 2009.

[107] R.R. Brooks and S.S. Iyengar. *Multi-sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall PTR, Upper Saddle River, NJ, 1997.

[108] W. Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference*, Portland, OR, USA, June 2005.

[109] W. Ren, R.W. Beard, and D.B. Kingston. Multi-agent kalman consensus with relative uncertainty. In *American Control Conference*, Portland, OR, USA, June 2005.

[110] B. Coifman, M. McCord, R.G. Mishalani, M. Iswalt, and Y. Ji. Roadway traffic monitoring from an unmanned aerial vehicle. *IEE Proceedings on Intelligent Transportation System*, 153(1):11, 2006.

[111] C.F. Lin and A.G. Ulsoy. Calculation of the time to lane crossing and analysis of its frequency distribution. In *American Control Conference*, 1995.

[112] C.-T. Lin, R.-C. Wu, S.-F. Liang, W.-H. Chao, Y.-J. Chen, and T.-P. Jung. Eeg-based drowsiness estimation for safety driving using independent component analysis. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 52(12):2726–2738, 2005.

[113] H. Cai, Y. Lin, and R.R. Mourant. Study on driver emotion in driver-vehicle-environment systems using multiple networked driving simulators. In *Proceedings of the Driving Simulation*, Iowa City, Sep 2007.

[114] Y. Lin, P. Tang, W.J. Zhang, and Q. Yu. Artificial neural network modelling of driver handling behaviour in a driver-vehicle-environment system. *International Journal of Vehicle Design*, 37(1):24–45, 2005.

[115] C.C. Loy, T. Xiang, and S. Gong. Modelling multi-object activity by gaussian processes. In *British Machine Vision Conference*, 2009.

[116] J. Will, L. Peel, and C. Claxton. Fast maritime anomaly detection using kd-tree gaussian processes. In *2nd IMA Mathematics in Defence*, Defence Academy, Shrivenham, UK, 2011.

[117] F. Johansson and G. Falkman. Detection of vessel anomalies - a bayesian network approach. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007.

[118] Y. Du, F. Chen, W. Xu, and Y. Li. Recognizing interaction activities using dynamic bayesian networks. In *ICPR*, 2006.

[119] S. Srivastava, K.K. Ng, and E.J. Delp. Co-ordinate mapping and analysis of vehicle trajectory for anomaly detection. In *IEEE International Conference on Multimedia and Expo (ICME)*, July 2011.

[120] X. Li, J. Han, S. Kim, and H. Gonzalez. ROAM: Rule- and motif-based anomaly detection in massive moving object data sets. In *7th SIAM International Conference on Data Mining*, 2007.

[121] R. Fraile and S.J. Maybank. Vehicle trajectory approximation and classification. In *British machine Vision Conference*, 1998.

[122] V.I. Shvetsov. Mathematical modeling of traffic flows. *Automation and Remote Control*, 64(11):1651–1689, 2003.

[123] P. Chakroborty. Models of vehicular traffic: An engineering perspective. *Physica A: Statistical Mechanics and Its Applications*, 372(1):151–161, 2006.

[124] R.W. Deming and L.I. Perlovsky. Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors. *Information Fusion*, 8:316–330, 2007.

[125] N. Oliver and A. P. Pentland. Graphical models for driver behaviour recognition in a smartcar. In *Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, US*, Oct. 2000.

[126] S. Kim, R.W. Zbikowski, A. Tsourdos, and B.A. White. Airborne monitoring of ground traffic behaviour for hidden threat assessment. In *13th International Conference on Information Fusion*, Endinburgh, UK, July 2010.

[127] P. Bonsall, R. Liu, and W. Young. Modeling safety-related driving behavior-impact of parameter values. *Transportation Research Part A*, 39(5):425–444, 2005.

[128] M. Junger, R. West, and R. Timman. Crime and risky behavior in traffic: An example of cross-situational consistency. *Journal of Research in Crime and Delinquency*, 38(4):439–459, 2001.

[129] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San diego, USA, 2006.

[130] D.J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964.

[131] E.H. Mamdani. Applications of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, 26(12):1182–1191, 1977.

[132] H. Hellendoorn D. Driankov and M. Reinfrank. *An Introduction to Fuzzy Control*. New York: Springer-Verlag, 1993.

[133] S. Martnez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[134] G. Gu, P. Chandler, C. Schumacher, A. Sparks, and M. Pachter. Optimal cooperative sensing using a teams of UAVs. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4):1446–1458, 2006.

[135] D.A. Lawrence. Lyapunov vector fields for UAV flock coordination. *2nd AIAA Unmanned Unlimited conference, workshop, and exhibit, Reston, VA, USA*, 2003.

[136] S. Morris and E.W. Frew. Cooperative tracking of moving targets by teams of autonomous unmanned air vehicles. Technical Report FA9550-04-C-0107, MLB Company, 2005.

[137] T.H. Summers, M.R. Akella, and M.J. Mears. Coordinated standoff tracking of moving targets: Control laws and information architectures. *Journal of Guidance, Control, and Dynamics*, 32(1):56–69, 2009.

[138] H. Chen, K. Chang, and C. Agate. Tracking with UAV using tangent-plus-lyapunov vector field guidance. *International Conference on Information Fusion, Seattle, WA, USA*, 2009.

[139] D. Kingston and R. Beard. UAV spaly state configuration for moving targets in wind. *Lecture Notes in Control and Information*, 369:109–128, 2007.

[140] R. Sepulchre, D.A. Paley, and N.E. Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52(5):811–824, 2007.

[141] D.A. Paley, N.E. Leonard, and R. Sepulchre. Stabilization of symmetric formations to motion around convex loops. *Systems and Control Letters*, 57:209–215, 2008.

[142] D.J. Klein and K.A. Morgansen. Controlled collective motion for trajectory tracking. In *American Control Conference*, American Control Conference, Minneapolis, Minnesota, 2006.

[143] C.G. Prevost, O. Theriault, A. Desbiens, and E. Poulin. Receding horizon model-based predictive control for dynamic target tracking: a comparative study. *AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois, USA*, 2009.

[144] H. Oh, S. Kim, H.S. Shin, B.A. White, A. Tsourdos, and C.A. Rabbath. Rendezvous and standoff target tracking guidance using differential geometry. *Journal of Intelligent and Robotic Systems*, 69:389–405, 2013.

[145] H.K. Khalil. *Nonlinear Systems (3rd Edition)*. Prentice Hall, 2002.

[146] M. Quigley, M.A. Goodrich, S. Griffiths, A. Eldredge, and R.W. Beard. Target acquisition, localisation, and surveillance using a fixed-wing mini-UAV and gimbaled camera. *IEEE international conference on robotics and automation, Barcelona, Spain*, 2005.

[147] J.E. Marsden and M. McCracken. *The Hopf Bifurcation and its applications, Vol 19 of Applied Mathematical Sciences*. Springer-Verlag, 1976.

[148] M. Krstic, I. Kanellakopoulos, P.V. Kokotovic, et al. *Nonlinear and adaptive control design*, volume 8. John Wiley & Sons New York, 1995.

[149] B.A. White, R. Zibkowski, and A. Tsourdos. Direct intercept guidance using differential geometry concepts. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):899–919, 2007.

[150] B.A. White, H.S. Shin, and A.Tsourdos. UAV obstacle avoidance using differential geometry concepts. In *18th IFAC World Congress*, Milano, Italy, Aug 2011.

[151] D.A. Lawrence, E.W. Frew, and W.J. Pisano. Lyapunov vector fields for autonomous unmanned aircraft flight control. *Journal of Guidance, Control, and Dynamics*, 31(5):1220–1229, 2008.

[152] B. O'Neill. *Elementary Differential Geometry (2nd ed.)*. San Diego, CA: Academic Press, 1997.

[153] J. Shin and H. Kim. Nonlinear model predictive formation flight. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems an Humans*, 39(5), 2009.

[154] H. Oh, S. Kim, A. Tsourdos, and B.A. White. Road-map assisted standoff tracking of moving ground vehicle using nonlinear model predictive control. In *American Control Conference*, Montreal, Canada, June 2012.

[155] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10:53–66, 1993.

[156] D.L. Massart R. De Maesschalck, D. Jouan-Rimbaud. The mahalanobisdistance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, 2000.

[157] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[158] M. Inaba, N. Katoh, and H. Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of 10th ACM Symposium on Computational Geometry*, pages 332–339, 1994.

[159] J.H Taylor. Cramer-rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Transactions on Automatic Control*, 24(2):343–344, 1979.

[160] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, Australian Centre for Field Robotics, Department of Aerospace, Mechatronic and Mechanical Engineering, The University of Sydney, 2002.

[161] T. Furukawa, F. Bourgault, B. Lavis, and H.F. Durrant-Whyte. Recursive bayesian search-and-tracking using coordinated UAVs for lost targets. In *IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006.

[162] A.Waldock and D. Corne. Multiple objective optimisation applied to route planning. In *Systems Engineering for Autonomous Systems (SEAS) - Defence Technology Centre (DTC) Conference*, Edinburgh, UK, July 2010.

[163] A. Waldock and D. Corne. *Exploiting Prior Information in Multi-objective Route Planning, Parallel Problem Solving from Nature - PPSN XII*, volume 7492 of *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2012.

[164] A. Waldock, B. Carse, and C. Melhuish. Hierarchical fuzzy rule based on systems using an information theoretic approach. *Soft Computing*, 10:867–879, 2006.

[165] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *International Conference on Computer Vision and Pattern Recognition*, 2005.

[166] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interactive multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34:103–123, 1998.

[167] C. Yang, M. Bakich, and E. Blasch. Nonlinear constrained tracking of targets on roads. *International Conference on Information Fusion*, 2005.

[168] C.E. Rasmussen and C.K.I. Williams. *Gaussian Process for Machine Learning*. MIT Press, 2006.

Appendices

Appendix A

Third and Fourth Case of Path Planning Using Constant Curvature Segments

The third case for the proposed path planning in Section 2.2 deals with the situation where the initial orientation is towards tangent intersection point while the final is opposite to that. From the geometry as shown in Fig. A.1, two different expressions for the angle θ can be obtained as:

$$\theta = 2\pi - \phi_s - \phi_f - \psi \tag{A.1}$$

$$\theta = 2\pi - \zeta_s - \zeta_f \tag{A.2}$$

where $\zeta_s = \frac{\pi}{2} - \phi_s$ and $\zeta_f = \frac{\pi}{2} - \phi_f$. Combing above equations gives:

$$\phi_f = \frac{\pi}{2} - \frac{\psi}{2} - \phi_s. \tag{A.3}$$

Since the direction of two arcs is the same, the summation of the turning angle should satisfy:

$$0 < \theta_s + \theta_f \leq \pi \tag{A.4}$$

leading to different boundary conditions from the one obtained in the first case as:

$$0 \leq \phi_s \leq \gamma_s \tag{A.5}$$

$$0 \leq \phi_f \leq \gamma_f. \tag{A.6}$$

Using Eq. (A.3) and Eq. (A.5), the final boundary of ϕ_s is given as:

$$\phi_s \geq \max[0, \alpha_{s_3}] \tag{A.7}$$

$$\phi_s \leq \min\left[\gamma_s, \frac{\pi}{2} - \frac{\psi}{2}\right] \tag{A.8}$$

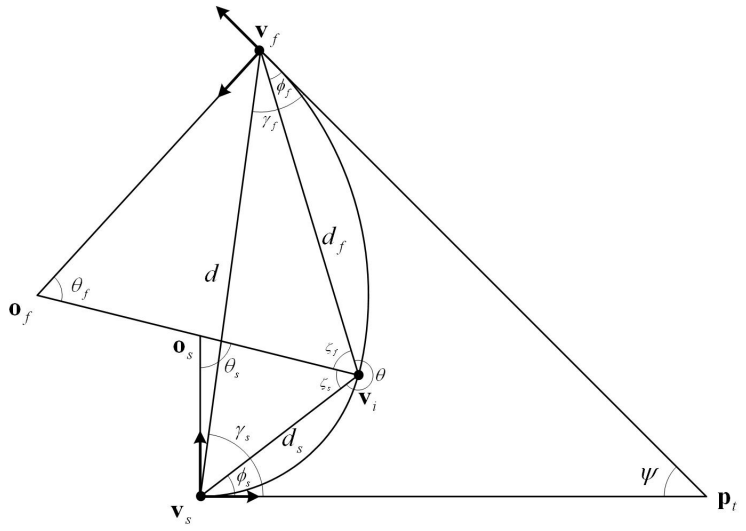


Figure A.1: Geometry for the third case

where $\alpha_{s_3} = \frac{\pi}{2} - \frac{\psi}{2} - \gamma_f$. Valid boundaries can be selected depending on required initial and final configurations. Considering the case where $\alpha_{s_3} = \frac{\pi}{2} - \frac{\psi}{2} - \gamma_f \leq 0$, the following relation holds:

$$\begin{aligned} \gamma_s &= \pi - \psi - \gamma_f \\ \gamma_s &\leq \pi - \psi - \frac{\pi}{2} + \frac{\psi}{2} = \frac{\pi}{2} - \frac{\psi}{2}. \end{aligned} \tag{A.9}$$

Similarly, the boundary for the case where $\alpha_{s_3} > 0$ can be obtained, and combing with Eq. (A.7), the final boundary of ϕ_s is obtained more specifically as:

$$\begin{aligned} &0 \leq \phi_s \leq \gamma_s, && \text{if } \alpha_{s_3} \leq 0 \\ &\tfrac{\pi}{2} - \tfrac{\psi}{2} - \gamma_f < \phi_s < \tfrac{\pi}{2} - \tfrac{\psi}{2}, && \text{otherwise.} \end{aligned} \tag{A.10}$$

The arc chord length d_s and d_f are obtained using triangle $(\mathbf{v}_s, \mathbf{v}_i, \mathbf{v}_f)$ and sine rule, resulting in the same equation as Eq. (2.32):

$$\begin{aligned} d_s &= d\frac{\sin(\gamma_f - \phi_f)}{\sin(2\pi - \theta)} = d\frac{\sin(\phi_f - \gamma_f)}{\sin(\theta)} \\ d_f &= d\frac{\sin(\gamma_s - \phi_s)}{\sin(2\pi - \theta)} = d\frac{\sin(\phi_s - \gamma_s)}{\sin(\theta)}. \end{aligned} \tag{A.11}$$

Other parameters like the curvature can be obtained using the same way as in the first case. Moreover, to address the physical constraints, the turning radius is first

expressed as a function of ϕ_s as done for the first case:

$$r_s = d\frac{\sin(\phi_f - \gamma_f)}{2\sin\theta\sin\phi_s} = d\frac{\sin(\phi_s - \alpha_{s_3})}{2\cos\frac{\psi}{2}\sin\phi_s}. \tag{A.12}$$

The characteristic of the turning radius of the initial arc for the third case is obtained as the following lemma.

Lemma A.0.1. *(Characteristic of the turning radius for the initial arc of 3rd case): For $0 < \psi < \pi$, $d > 0$ and $0 \leq \phi_s < \pi$, initial turning radius r_s is monotonically decreasing with increasing ϕ_s for $\alpha_{s_3} < 0$, and increasing for $\alpha_{s_3} > 0$ in the third case.*

Proof. Differentiating Eq. (A.12) with respect to ϕ_s gives:

$$\frac{\partial r_s}{\partial \phi_s} = \frac{d}{2\cos\theta}\frac{\cos(\phi_s - \alpha_{s_3})\sin\phi_s - \cos\phi_s\sin(\phi_s - \alpha_{s_3})}{\sin^2\phi_s}. \tag{A.13}$$

In case that $\alpha_{s_3} < 0$, since $\phi_s \in (0, \gamma_s)$ from Eq. (A.10), the following holds:

$$(\phi_s - \alpha_{s_3})_{inf} = -\alpha_{s_3} > 0 \tag{A.14}$$

$$(\phi_s - \alpha_{s_3})_{sup} = \gamma_s - \frac{\pi}{2} + \frac{\psi}{2} + \gamma_f < \frac{\pi}{2}. \tag{A.15}$$

When $\alpha_{s_3} > 0$, since $\phi_s \in (\frac{\pi}{2} - \frac{\psi}{2} - \gamma_f, \frac{\pi}{2} - \frac{\psi}{2})$ from Eq. (A.10), the boundary of $\phi_s - \alpha_{s_3}$ is obtained as:

$$(\phi_s - \alpha_{s_3})_{inf} = 0 \tag{A.16}$$

$$(\phi_s - \alpha_{s_3})_{sup} = \gamma_f \tag{A.17}$$

which implies $(\phi_s - \alpha_{s_3}) \in (0, \pi)$. For $d > 0$ and $\psi < \pi$, the sign of Eq. (A.13) depends on its second numerator as:

$$\begin{aligned} & \cos(\phi_s - \alpha_{s_3})\sin\phi_s - \cos\phi_s\sin(\phi_s - \alpha_{s_3}) \\ = \; & \sin\phi_s\sin(\phi_s - \alpha_{s_3})(\cot(\phi_s - \alpha_{s_3}) - \cot\phi_s). \end{aligned} \tag{A.18}$$

For $\alpha_{s_3} < 0$, by recalling that $\phi_s \in (0, \pi)$, $(\phi_s - \alpha_{s_3}) \in (0, \frac{\pi}{2})$, and the cotangent function is monotonically decreasing in such an interval, it can be concluded that the sign of Eq. (A.14) is negative since $\phi_s < \phi_s - \alpha_{s_3}$. Similarly, for $\alpha_{s_3} > 0$, since $\phi_s > \phi_s - \alpha_{s_3}$, the sign of Eq. (A.14) is positive, resulting in:

$$\frac{\partial r_s}{\partial \phi_s} < 0, \text{if } \alpha_{s_3} < 0 \tag{A.19}$$

$$\frac{\partial r_s}{\partial \phi_s} > 0, \text{if } \alpha_{s_3} > 0 \tag{A.20}$$

□

Once monotonically increasing or decreasing behavior is determined, the boundary of ϕ_s for the feasible path can be obtained by imposing the following condition:

$$\begin{aligned} r_s = d\frac{\sin(\phi_s - \alpha_{s_3})}{2\cos\frac{\psi}{2}\sin\phi_s} &\geq r_m \\ \frac{\sin\phi_s\cos\alpha_{s_3} - \cos\phi_s\sin\alpha_{s_3}}{\sin\phi_s} &\geq \frac{2r_m\cos\frac{\psi}{2}}{d} \\ -\cot\phi_s\sin\alpha_{s_3} &\geq \frac{2r_m\cos\frac{\psi}{2}}{d} - \cos\alpha_{s_3}. \end{aligned} \tag{A.21}$$

Depending on the sign of α_{s_3}, the feasible bounds on ϕ_s are obtained as:

$$\begin{aligned} &\phi_s \leq \tan^{-1}\left[\frac{1}{\beta_{s_3} + \cot\alpha_{s_3}}\right], \text{ if } \alpha_{s_3} \leq 0 \\ &\phi_s > \tan^{-1}\left[\frac{1}{\beta_{s_3} + \cot\alpha_{s_3}}\right], \text{if } \alpha_{s_3} > 0 \end{aligned} \tag{A.22}$$

where

$$\beta_{s_3} = -\frac{2r_m\cos\frac{\psi}{2}}{d\sin\alpha_{s_3}}.$$

Similar steps can be done for the final arc. Using $\alpha_{s_3} = \frac{\pi}{2} - \frac{\psi}{2} - \gamma_f = -\frac{\pi}{2} + \frac{\psi}{2} + \gamma_s$ and Eq. (A.3) gives:

$$\frac{\partial r_f}{\partial\phi_s} < 0, \text{if } \alpha_{s_3} < 0 \text{ or } \alpha_{f_3} > 0 \tag{A.23}$$

$$\frac{\partial r_f}{\partial\phi_s} > 0, \text{if } \alpha_{s_3} > 0 \text{ or } \alpha_{f_3} < 0 \tag{A.24}$$

where $\alpha_{f_3} = \frac{\pi}{2} - \frac{\psi}{2} - \gamma_s$, and another upper boundary for ϕ_s as:

$$\begin{aligned} &\phi_s \leq \frac{\pi}{2} - \frac{\psi}{2} - \tan^{-1}\left[\frac{1}{\beta_{f_3} + \cot\alpha_{f_3}}\right], \text{ if } \alpha_{s_3} \leq 0 \\ &\phi_s > \frac{\pi}{2} - \frac{\psi}{2} - \tan^{-1}\left[\frac{1}{\beta_{f_3} + \cot\alpha f_3}\right], \text{ otherwise} \end{aligned} \tag{A.25}$$

where

$$\beta_{f_3} = -\frac{2r_m\cos\frac{\psi}{2}}{d\cos\left(\frac{\pi}{2} + \gamma_s\right)}.$$

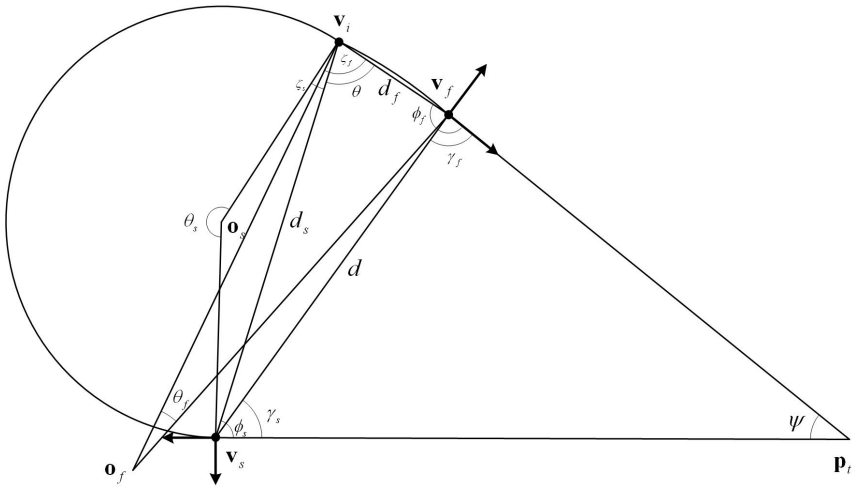


Figure A.2: Geometry for the fourth case

Note that for $\alpha_{s_3} \leq 0$, since both r_s and r_f decrease with increasing ϕ_s from Eq. (A.19) and Eq. (A.23), only upper bound on ϕ_s can be obtained using turning radius constraints for the initial and final arc; for $\alpha_{s_3} \geq 0$, only lower bound can be computed. Therefore, by choosing the appropriate ϕ_s using the boundary conditions from Eq. (A.10), Eq. (A.22), Eq. (A.25) and corresponding ϕ_f from Eq. (A.3), the flyable path for the UAV is generated.

The fourth case deals with the situation where the initial orientation is opposite to tangent intersection point while the final is towards that. An example of such a configuration is given in Fig. A.2. A feasible solution requires:

$$\pi < \theta_s + \theta_f \leq 2\pi \tag{A.26}$$

which gives the boundary of ϕ_s using a similar way as explained for the third case as:

$$\begin{array}{ll} \gamma_s \leq \phi_s \leq \pi, & \text{if } \alpha_{s_4} \leq 0 \\ \frac{\pi}{2} - \frac{\psi}{2} < \phi_s < \frac{3}{2}\pi - \frac{\psi}{2} - \gamma_f, & \text{otherwise} \end{array} \tag{A.27}$$

where $\alpha_{s_4} = \gamma_s - \frac{\pi}{2} + \frac{\psi}{2}$. Similarly, the feasible bounds on ϕ_s from initial arc condition can be obtained as:

$$\begin{aligned} \phi_s &\leq \tan^{-1}\left[\frac{1}{\beta_{s_4} + \tan\left(\frac{\psi}{2} + \gamma_f\right)}\right], \text{ if } \alpha_{s_4} \leq 0 \\ \phi_s &> \tan^{-1}\left[\frac{1}{\beta_{s_4} + \tan\left(\frac{\psi}{2} + \gamma_f\right)}\right], \text{ otherwise} \end{aligned} \tag{A.28}$$

where

$$\beta_{s_4} = -\frac{2r_m \cos\frac{\psi}{2}}{d\cos\left(\frac{\psi}{2}+\gamma_f\right)}.$$

The other feasible bound on ϕ_s from the final arc condition is obtained as:

$$\begin{aligned}
\phi_s &\leq \frac{3}{2}\pi - \frac{\psi}{2} - \tan^{-1}\left[\frac{1}{\beta_{s_4}+\tan\left(\frac{\psi}{2}+\gamma_s\right)}\right], \text{ if } \alpha_{s_4} \leq 0 \\
\phi_s &> \frac{3}{2}\pi - \frac{\psi}{2} - \tan^{-1}\left[\frac{1}{\beta_{s_4}+\tan\left(\frac{\psi}{2}+\gamma_s\right)}\right], \text{ otherwise}
\end{aligned} \tag{A.29}$$

where

$$\beta_{f_4} = -\frac{2r_m \cos\frac{\psi}{2}}{d\cos\left(\frac{\psi}{2}+\gamma_s\right)}.$$

By choosing the appropriate ϕ_s using the boundary conditions from Eq. (A.27), Eq. (A.28) and Eq. (A.29), the flyable path for the UAV is generated.

Appendix B

Anomaly Detection Using GP and Road Map

This appendix presents an airborne behaviour monitoring methodology using both a learning approach based on Gaussian processes and a domain knowledge approach provided from road map information, in relation to Chapter 5. For learning approach for anomaly detection, many works have been performed using parametric models such as Bayesian network [117, 118] and hidden Markov models [165]. In order to alleviate the complexity of the problem, Gaussian process (GP) models are also applied. Will *et al.* [116] used the Kd-tree GPs to model shipping behaviour and maritime anomaly detection. Loy *et al.* [115] also applied the GPs for activity modelling and real-time anomaly detection of a ground traffic flow using surveillance video footage. Compared to commonly used parametric models since GPs are not a parametric model, users do not need to be concerned about if it is possible for the model to fit the data [115]. The use of a flexible, non-parametric model alleviates the difficulty of selecting appropriate model complexity encountered in parametric models. GP models need fewer parameters; they are thus less likely to overfit given sparse data of moving vehicles.

The velocity profile of the vehicle with respect to its position is one of principal factors which provides some measure of suspicious or abnormal behaviour. Therefore, the Gaussian process regression algorithm is applied to the velocity profile along the one-dimensional representation of the vehicle position. Since some vehicles can move over off-road terrain to avoid a police checkpoint or to closely monitor a particular place, temporal probability of on/off-road modes is another important source for abnormal behaviour detection. In order to obtain this mode probability whilst enabling monitoring of moving ground target, an interactive multiple model (IMM) filter [166] is applied. The proposed IMM filter comprises an on-road moving mode using a road-constrained filter and an off-road moving mode using a conventional filter so that both on and off mode probabilities (which are complementary to each other) can be obtained. In this

study, in order to combine these two approaches, a novel hybrid algorithm is also developed based on joint probability theory.

The key innovation of the proposed approach lies in the hybrid algorithm which integrates the two different philosophies and inherently brings several advantages. The proposed algorithm does not require the definition of specific behaviours in order to detect suspicious behaviour. Exploiting the velocity profile of a vehicle on a specific road from statistical traffic data allows consideration of road conditions such as slope, roughness or curve. Moreover, anomaly measure on the vehicle behaviours can be obtained by evaluating the quality of predictions from the deviation of the actual target data using the predictive distribution. Since these anomaly scores are augmented with the on/off-road mode probability, the proposed algorithm can also complement the weakness of relying on the learning data which is difficult to incorporate a certain anomalous behaviour aspect.

The overall structure of this section is given as follows. Section B.1 explains a road-constrained IMM filter design. Section B.2 introduces the 1-D representation of the position on the road of interest, and proposes a novel anomalous behaviour detection scheme combining predictive log-likelihood from GP regression and the off-road mode probability. Section B.4 presents numerical simulation results of behaviour monitoring scenario using realistic ground vehicle trajectory data. The overall procedure flow chart of the behaviour monitoring of ground vehicles using UAVs is shown in Fig. B.1.

B.1 Road-Constrained IMM Filter

Although a normal vehicle tends to move along the road only, some of vehicles can move onto off-road terrain for some reasons, e.g. to avoid a police checkpoint or to closely monitor a particular place. Therefore, an estimation algorithm should be able to recognise both off-road and on-road movement depending on the situation, rather than constraining the estimates to the roadmap. To address this, an IMM filter [166] is applied which combines an off-road mode using the conventional filter and an on-road mode using a road-constrained filter as explained in Section 3.4.

Let $M_k = \{0, 1, \cdots, N_k - 1\}$ be the mode set of the IMM estimator in the interval

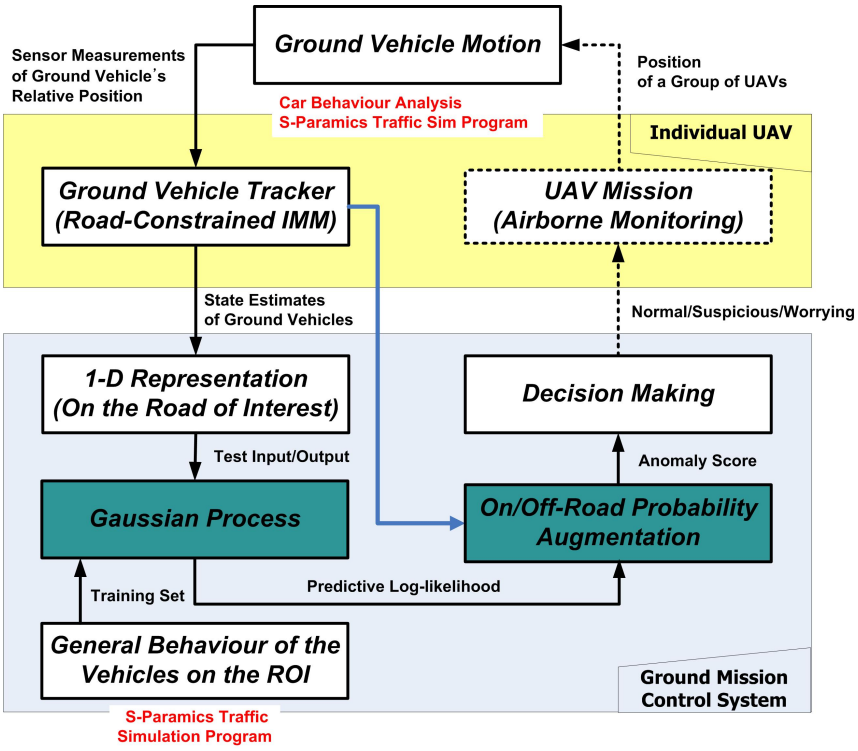


Figure B.1: An overall flow chart of behaviour monitoring

$(k-1,k]$ and $\mu_{j,k}$ be the probability that the mode is active during the time interval so that:

$$\mu_{j,k} = p\{m_k = j \in M_k | \mathbf{Z}_k\} \tag{B.1}$$

where $p\{\cdot\}$ denotes the probability, and $\mathbf{Z}_k$ is the measurement sequence over time k. Physically, m_k represents the type of motion (off-road $m_k = 0$ or on-road $m_k = m \in \{1, 2, \cdots, N_k - 1\}$), and this is modelled as a Markov process with a first-order homogeneous Markovian transition of modes:

$$\pi_{ij,k} = p\{m_k = j | m_{k-1} = i\} \quad \forall\, i, j \in M_k \tag{B.2}$$

where π_{ij} is the Markov transition probability from mode i to mode j.

The IMM filter uses a recursive algorithm composed of the following four processes: interaction/mixing, filtering, model probability update, and estimate combination. The input to a filter matching a certain mode is obtained by mixing the state estimates, under the assumption that this particular mode is active at the present step. Then, a conventional filtering is performed in parallel. The model probabilities are updated based on model-conditional likelihood functions.

Finally, overall states are estimated from the probabilistically weighted sum of the outputs from each filter.

B.1.1 Interaction/mixing process

A predicted mode probability is obtained using the transition probability as:

$$\mu_{j,k|k-1} = \sum_{i \in M_k} \pi_{ij,k} \mu_{i,k-1} \; \forall \; j \in M_k. \tag{B.3}$$

Then, a mixing probability is given as:

$$\mu_{i|j,k-1} = \frac{\pi_{ij,k-1} \mu_{i,k-1}}{\mu_{j,k|k-1}} \; \forall \; i, j \in M_k. \tag{B.4}$$

The initial value of the state and covariance for the next step can be obtained using mixing probability and estimates from the previous step to give:

$$\hat{\mathbf{x}}^{0,t}_{j,k-1|k-1} = \sum_{i \in M_k} \hat{\mathbf{x}}^{t}_{k-1|k-1} \mu_{i|j,k-1} \tag{B.5}$$

$$\begin{aligned} P^{0}_{j,k-1|k-1} = \sum_{i \in M_k} \{ & P_{i,k-1|k-1} + [\hat{\mathbf{x}}^{0,t}_{j,k-1|k-1} - \hat{\mathbf{x}}^{t}_{i,k-1|k-1}] \\ & \times [\hat{\mathbf{x}}^{0,t}_{j,k-1|k-1} - \hat{\mathbf{x}}^{t}_{i,k-1|k-1}]^T \} \mu_{i|j,k-1} \; \forall \; j \in M_k. \end{aligned} \tag{B.6}$$

B.1.2 Mode-conditioned filtering process

The EKF, with the corresponding models (off-road or on-road), is used as the individual mode-conditioned filter providing the updated state $\hat{\mathbf{x}}^{t}_{j,k|k}$ and covariance $P_{j,k|k}$.

B.1.3 Model Probability Update Process

Model probability offers an important source for abnormal behaviour detection, which is discussed in the next section. For this, a likelihood function is defined using the following general Gaussian function:

$$L_{j,k} = \frac{1}{\sqrt{|2\pi S_{j,k}|}} \exp\left[-\frac{1}{2} v^{T}_{j,k} S^{-1}_{j,k} v_{j,k} \right] \tag{B.7}$$

where $S_{j,k}$ and $v_{j,k}$ are the residuals of mode j and its covariance, as defined in Eq. (4.37) and Eq. (4.38), respectively. Then, the model probability is updated by using:

$$\mu_{j,k} = \frac{\mu_{j,k|k-1} L_{j,k}}{\sum_{i \in M_k} \mu_{i,k|k-1} L_{i,k}} \; \forall \; j \in M_k. \tag{B.8}$$

B.1.4 Estimate combination process

The final estimate can be computed using the weighted sum of each filter's estimate, using the model probability as:

$$\hat{\mathbf{x}}^t_{k|k} = \sum_{j \in M_k} \hat{\mathbf{x}}^t_{j,k|k} \mu_{j,k}. \tag{B.9}$$

The estimated states from the combination process along with the model probability will be used, not only for tracking of ground vehicles, but also for behaviour recognition in the following section.

B.2 Behaviour Monitoring

This section firstly introduces a one-dimensional (1-D) representation of the position of the vehicle on the road to simplify the vehicle position. Then a novel anomalous behaviour detection scheme is proposed, using predictive log-likelihood from a Gaussian process regression, as well as the off-road mode probability.

B.2.1 1-D representation of the position

To represent the behaviour of the ground vehicle travelling on the road, a 1-D representation of the vehicle's position is introduced, using the similar approach to [167]. The ground target trajectory, when in traverse of a road-network, can be modelled using curved segments connected with arcs of different curvature, and a moving target can be located uniquely using the time, street name (a road segment), and street number (distance relative to a starting point on the street). This study assumes a width-less road since the road width is negligible considering

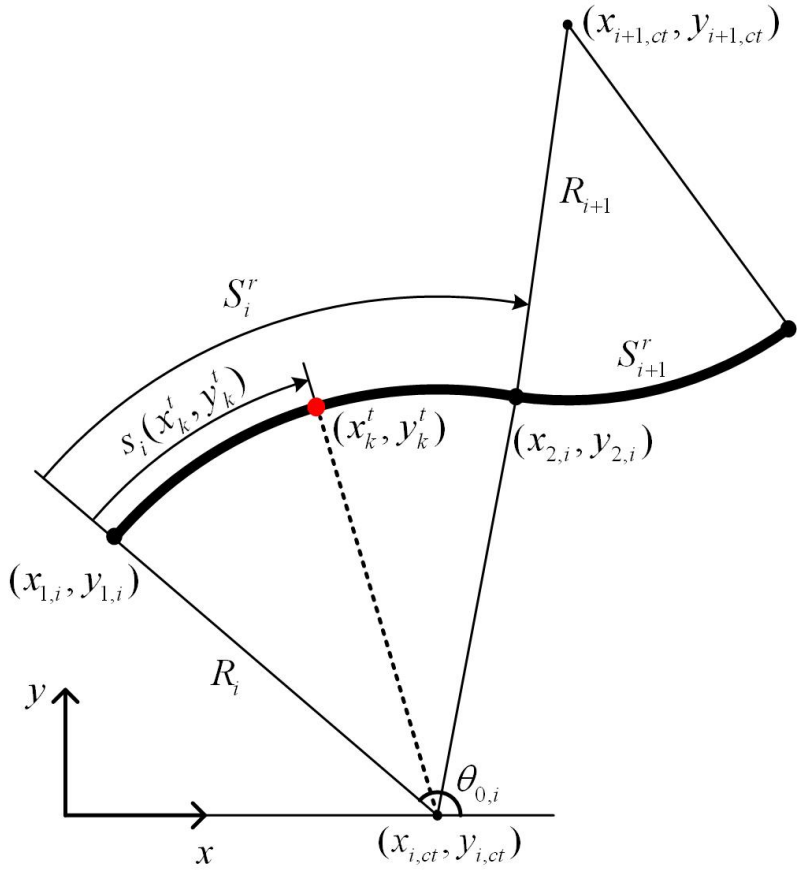


Figure B.2: 1-D representation of road segments

the resolution and error of the sensor onboard the UAV. Let us consider a circular arc characterised by a starting point $(x_{1,i}, y_{1,i})$, an end point $(x_{2,i}, y_{2,i})$, the centre of the arc $(x_{i,ct}, y_{i,ct})$, and the arc curvature κ_i for the i^{th} road (which can be obtained from the road approximation algorithm), as shown in Fig. B.2. The location of the vehicle is determined by its distance from a starting point which will be referred to as the mileage count [167]. Assuming multiple n_r roads are connected in the region of interest, a 2-D point (x, y) on the curved road can be expressed using the mileage count s_i as:

$$x(s_i) = x_{i,ct} + R_i \cos\left(\theta_{0,i} + \frac{s_i}{R_i}\right) \tag{B.10}$$

$$y(s_i) = y_{i,ct} + R_i \sin\left(\theta_{0,i} + \frac{s_i}{R_i}\right) \tag{B.11}$$

where $0 \leq s_i \leq S_i^r$, $R_i = 1/\kappa_i$, and $\theta_{0,i}$ and S_i^r are given by:

$$\theta_{0,i} = \tan^{-1}\left(\frac{y_{1,i} - y_{i,ct}}{x_{1,i} - x_{i,ct}}\right) \tag{B.12}$$

$$S_i^r = R_i \cos^{-1}\left[1 - \frac{(x_{2,i} - x_{1,i})^2 + (y_{2,i} - y_{1,i})^2}{2R_i^2}\right]. \tag{B.13}$$

The mileage s_i of the ground vehicle at the position (x_k^t, y_k^t) can be obtained using Eq. (B.10) inversely, as:

$$\begin{aligned} s_i(x_k^t, y_k^t) &= R_i\left[\cos^{-1}\left(\frac{x_k^t - x_{i,ct}}{R_i}\right) - \theta_{0,i}\right] \\ &= R_i\left[\sin^{-1}\left(\frac{y_k^t - y_{i,ct}}{R_i}\right) - \theta_{0,i}\right]. \end{aligned} \tag{B.14}$$

A normalised mileage count $\bar{s}$ of the vehicle position (x_k^t, y_k^t) on the i^{th} road, among n_r roads of interest, can be obtained as:

$$\bar{s}(x_k^t, y_k^t) = \frac{\left(\sum_{j=1}^{i-1} S_j^r\right) + s_i(x_k^t, y_k^t)}{\sum_{j=1}^{n_r} S_j^r}. \tag{B.15}$$

Note that the road on which the vehicle is travelling can be determined by the road-constrained IMM filter. Even in the case that the position of the vehicle is not exactly on the road, it can be constrained to be on the road by using the road-constrained filter, as explained in the previous section. Using this 1-D representation, the behaviour of the vehicle on the road can simply be expressed as its velocity and acceleration along with other quantities, at a specific position $\bar{s} \in [0, 1]$ on the road of interest.

B.3 Gaussian process behaviour modelling

A Gaussian process (GP) is used to model the behaviour of the vehicle. The GP can be viewed as a stochastic process or a Gaussian distribution over functions. The GPs are used to infer or predict function values at a finite set of test points using the observed data. Since GP is not a parametric model, users do not need to be concerned about whether it is possible for the model to fit the data or not. The GP for a regression problem [168] is briefly explained in the following.

First of all, a standard regression model is defined as $y_{GP} = f(\mathbf{x}) + \epsilon$, where $\mathbf{x}$ is an input vector (covariates) of dimension D, and y_{GP} is a scalar output (dependent variable). The noise ϵ is assumed to be an independent and identically distributed Gaussian distribution, with zero mean and variance σ^2. Then, Gaussian process

$f(\mathbf{x})$ is specified by its mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and the covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$. Since this study assumes a zero-mean GP, the process can be expressed as $f(\mathbf{x}) \sim GP(0, k(\mathbf{x}, \mathbf{x}'))$. A training set with N_o observation is expressed as $\mathcal{D} = \{(\mathbf{x}_n, y_{GP,n}) | n = 1, \cdots, N_o\} = \{\mathbf{X}, \mathbf{y}_{GP}\}$, and the following squared exponential covariance function is used:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \sum (\mathbf{x} - \mathbf{x}')\right) \tag{B.16}$$

where σ_f determines the magnitude, and $\sum = l^{-2}I$ is an isotropic covariance function.

Given a GP model with the covariance function defined as above, the fitness of this model to the training set can be evaluated using the marginal likelihood conditioned on the hyper-parameters θ (which define the covariance function and the Gaussian noise) in the form:

$$\log p(\mathbf{y}_{GP}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}_{GP}^\top \mathbf{K}^{-1}\mathbf{y}_{GP} - \frac{1}{2}\log|\mathbf{K}| - \frac{N_o}{2} \tag{B.17}$$

where $\mathbf{K}$ denotes the matrix of the covariances of all pairs of training points with $\mathbf{K}_{ij} = k(x_i, x_j)$. The hyper-parameters are optimised to provide good predictions, using the partial derivatives of Eq. (B.17) with respect to the hyper-parameter and by using a conjugate gradient optimiser.

Given a training set $\mathcal{D}$ and a test input vector $\mathbf{x}_*$, the predictive distribution for Gaussian process regression is computed by conditioning the joint Gaussian prior distribution of the observed output values $\mathbf{y}_{GP}$ and the function values f_* at the test locations on $\mathbf{X}$ and $\mathbf{x}_*$ to give:

$$f_*|\mathbf{X}, \mathbf{y}_{GP}, \mathbf{x}_* \sim \mathcal{N}\left(\bar{f_*}, \mathbb{V}[f_*]\right) \tag{B.18}$$

where the mean and variance are defined as:

$$\bar{f_*} = \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 I)^{-1}\mathbf{y}_{GP} \tag{B.19}$$

$$\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 I)^{-1}\mathbf{k}_* \tag{B.20}$$

where $\mathbf{k}_*$ denotes the vector of covariance between the test and the training points.

The output y_{GP} of the GP model is the speed of the vehicle ($v^t = \sqrt{(\dot{x}^t)^2 + (\dot{y}^t)^2}$) observed at positions on the road $\bar{s}(x^t, y^t) \in [0, 1]$. It is assumed that the training

data set is available using the S-Paramics traffic simulation software [94] in advance, and the test inputs and outputs come from the road-constrained IMM filter, with sensor measurements on the UAV monitoring several ground vehicles in real time. Note that, although the 1-D position $\bar{s}$ is used as an input variable currently, it can be extended to a vector including other relevant parameters such as time zone (day/night or weekday/weekend) or environmental factors (congestion or accidents) to more closely capture real world characteristics. One of limitations for using this approach is that training data should be given for the road (or region). However, if some of the road is important and under surveillance, traffic data could be readily available in advance or can be collected in real-time. Another limitation is that it is hard to determine a crisp boundary between normal and anomalous behaviour, whereas classification based on supervised learning starts from a known value or definition for each class.

B.3.1 Hybrid anomalous behaviour detection

Given the prediction method, one can evaluate the quality of predictions measuring the deviation of the actual test data from the predictive distribution. This quality can be directly related to anomalous behaviour detection, since it provides how normally or abnormally the test vehicle behaves, compared to the general vehicles. The simplest and intuitive way of measuring the quality of fit is to compute the squared residual $SR_k = (v_k^t - \bar{f}(\bar{s}(x_k^t, y_k^t)))^2$ between the mean prediction and the test output at each test point. To consider the predictive uncertainty additionally, a more conservative anomaly score in the form of negative log probability of the test output v_k^t can be used as [168]:

$$-\log p(v_k^t|\mathcal{D}, \bar{s}) = \frac{1}{2}\log(2\pi\sigma_*^2) + \frac{(v_k^t - \bar{f}(\bar{s}))^2}{2\sigma_*^2} \tag{B.21}$$

where the predictive variance for the GP regression is computed as $\sigma_*^2 = \mathbb{V}[f_*] + \sigma_n^2$. The noise variance σ_n^2 is included, since we are predicting the noisy output. Note that the more accurate the prediction of GP model is and the larger the predictive variance is, the less score Eq. (B.21) results. This occurs when either the prediction is correct (i.e. $v_k^t - \bar{f}(\bar{s}) \simeq 0$, or equivalently, the test data fits the normal behaviour obtained from the training data) or the model is uncertain about prediction with limited training data.

The log predictive probability in Eq. (B.21) can provide only a partial knowledge of the behaviour based on the velocity profile. In order to make a reliable final decision of abnormal behaviour, other critical sources to recognise the abnormal behaviour need to be considered. Since the mode probability $\mu_{j,k}$ of the vehicle moving off-road terrain ($j = 0$) or on the roads of interest ($j = 1, 2, \cdots, N_k - 1$) is one of most important parameters for the abnormal behaviour recognition, this study proposes a novel hybrid behaviour recognition algorithm using not only a Gaussian process but also the mode probability.

Assuming that the probability from the GP regression and mode probability are independent to each other (that is, the occurrence of one does not change the probability of the other occurring), an augmented abnormal score can be defined using the negative log of the joint probability, which is the product of the probabilities of each occurring as:

$$\begin{aligned} score_k &= -\log\left[p(v_k^t|\mathcal{D}, \bar{s}) \times \left(\sum_{j=1}^{N_k-1} \mu_{j,k}\right)^{\gamma}\right] \\ &= -\log\left[p(v_k^t|\mathcal{D}, \bar{s}) \times (1 - \mu_{0,k})^{\gamma}\right] \\ &= -\log p(v_k^t|\mathcal{D}, \bar{s}) - \gamma \log(1 - \mu_{0,k}) \end{aligned} \tag{B.22}$$

where γ is a weighting factor, controlling the relative importance between the GP predictive and the on-road mode probability. From Eq. (B.22), it is clear that large values of the score indicate that abnormal behaviour is likely to be occurring. If one of aforementioned probabilities (GP regression and on-road mode) is small, an augment abnormal score has a large value (c.f. Eq. (B.22)). As both probabilities become small, since the product of them makes the joint probability smaller, an augment abnormal score gets bigger, which implies that the vehicle behaviour is likely to be suspicious or abnormal.

Note that an abnormal velocity does not necessarily cause off-road driving, whereas the deviation of the vehicle from the road can lead to an abnormal velocity caused by change of surface condition or traffic flow. Therefore, there is a need to investigate the conditional probability or dependence of those two events, rather than an independence assumption, as used in Eq. (B.22). In order to detect abnormal behaviour, a certain detection threshold needs to be determined by considering the success and false alarm rates, defined by the sensor characteristics and environment. These aspects remain for future work.

B.4 Numerical Simulations

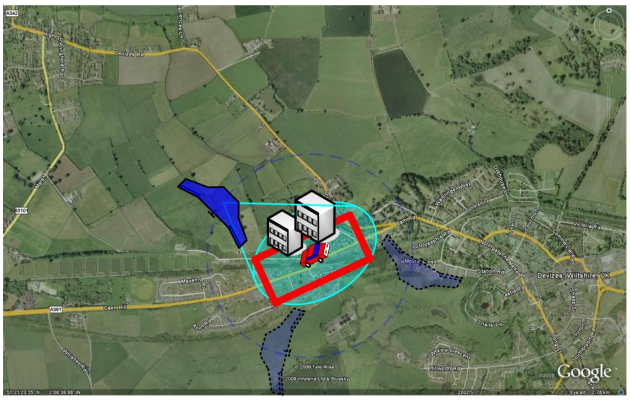
This section carries out a numerical simulation using the proposed anomalous behaviour detection algorithm for a moving ground target using a UAV loitering around the region of interest. The ground target trajectory is obtained from the S-Paramics [94] traffic model of Devizes at 2 Hz. It is used to generate GMTI measurements, composed of relative range and azimuth angle, having white noise with standard deviation $(\sigma_r, \sigma_\phi) = (7\ m, 2\ deg)$. Training data for the GP regression is also obtained from the S-Paramics model for 15 vehicles passing through the region of interest, as shown in Fig. B.3(a). Additionally, it is assumed that there is only one road (but composed of multiple segments) around the region of interest. In other words, we used a fixed mode set $M_k = \{0, 1\}$ consisting of two measurement models, with common acceleration dynamics of the ground vehicle for the IMM filter: one is a conventional model for a vehicle moving on off-road terrain, as described in Eq. (4.16), and the other is a pseudo-measurement augmented model moving on the road of interest, as described in Eq. (4.36). The initial mode probability is set to $(\mu_0, \mu_1) = (0.2, 0.8)$, and the Markov transition probability is given as:

$$\begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix} \tag{B.23}$$

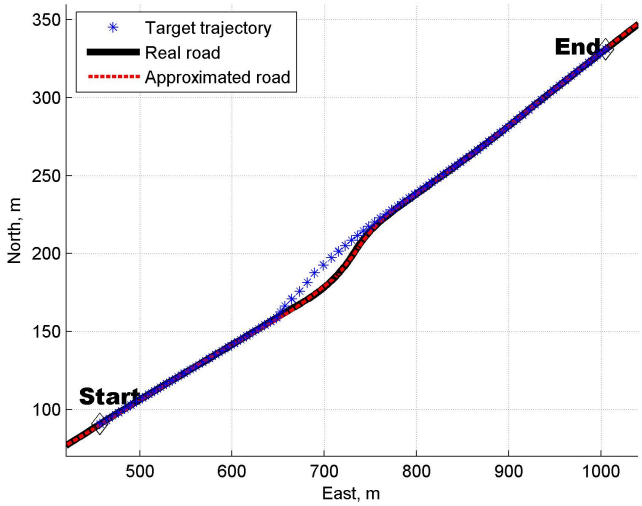
Note that the number of models can be changed to consider multiple roads at an intersection, using a variable-structure IMM concept [32, 90].

Figure B.3(b) shows the real roads of interest, their approximation, and the target trajectory from the S-Paramics simulation, with a slight change of the route (on-road → de-tour to avoid a police check point or to monitor a certain facility more closely → on-road). Figures B.4(a)~B.4(b) show the Gaussian process regression with a training set and test data from the road-constrained IMM filter of the vehicle having normal speed, and the on/off-road mode probability. Depending on the similarity of the velocity profile of the test ground vehicle and the training mean value from the GP, the predictive log-likelihood varies as shown by the blue line in Fig. B.4(c). In the figure, threshold of 30 is selected as an example in order to show that using predictive log likelihood can fail to detect anomalous behaviour as it relies only on velocity information. By contrast, since the off-road

mode probability is high when the vehicle is performing a detour from the road, the modified anomaly score from Eq. (B.22) is higher than the threshold. Note that this threshold value needs to be tuned with a weighting factor γ depending on the situations. Figure B.5 shows the simulation results for the same situation as the previous one but with the vehicle having abnormal speed during detour. In this case, since the speed of the test vehicle is quite different from the training data, as well as off-road mode probability being high (around $\bar{s} = 0.6$), the anomaly score shows a much larger value than the previous case, demonstrating that the anomalous or suspicious behaviour is highly likely to occur at that position.

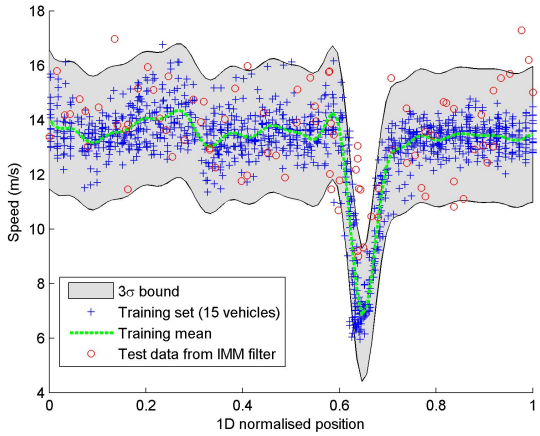


(a) Google map with region of interest

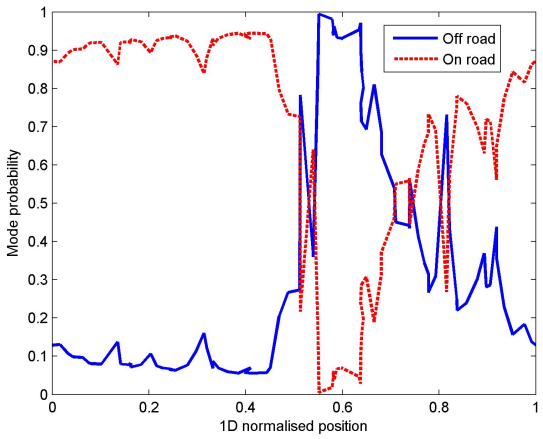


(b) Road approximation and target trajectory

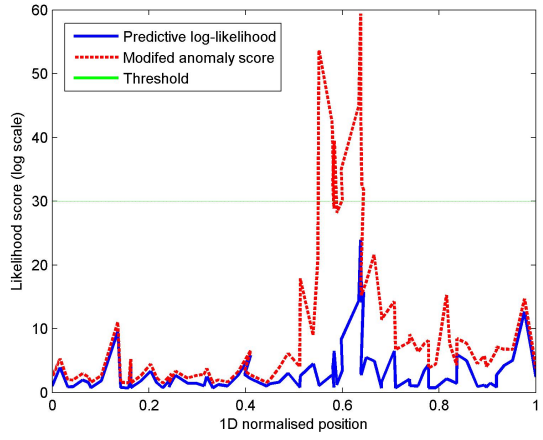
Figure B.3: Road approximation and target trajectory in the region of interest



(a) Gaussian process regression

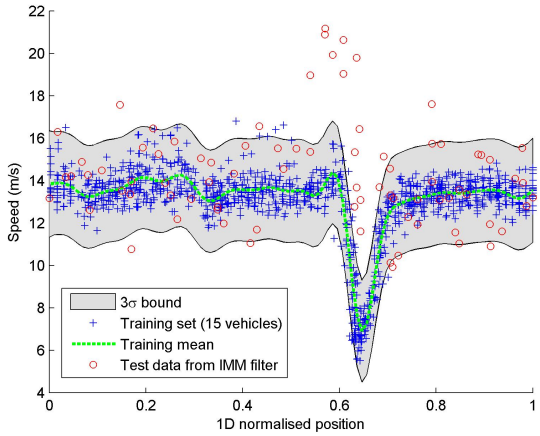


(b) On/off-road mode probability from IMM filter

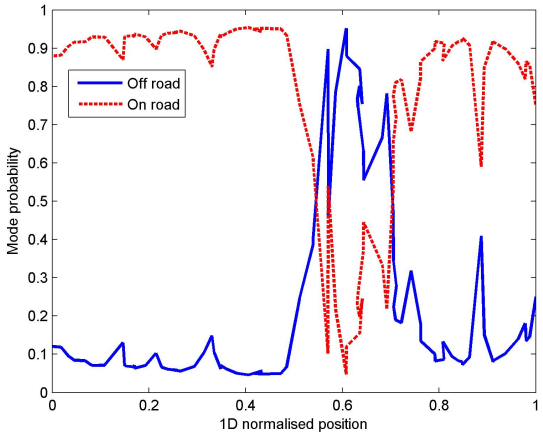


(c) Predictive log-likelihood and modified anomaly score ($\gamma = 10$)

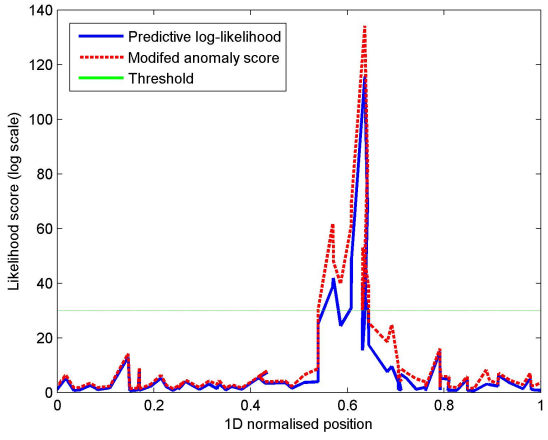
Figure B.4: Behaviour monitoring result with training set (15 vehicles from S-Paramics traffic simulator) and test data from road-constrained IMM filter with normal speed



(a) Gaussian process regression



(b) On/off-road mode probability from IMM filter



(c) Predictive log-likelihood and modified anomaly score ($\gamma = 10$)

Figure B.5: Behaviour monitoring result with training set (15 vehicles from S-Paramics traffic simulator) and test data from road-constrained IMM filter with abnormal speed